PHYSICS

Advancing FaaS Applications in the Cloud Continuum

Presentation for Red Hat Research Days -- Nov 16, 2022

Main Speaker: Georgios Kousiouris (HUA)
Conversation Leader: Luis Tomás Bolívar (RHT) and Yiannis Georgiou (RYAX)

This project has received funding from the European Union's horizon 2020
research and innovation programme under grant agreement no 101017047

AGENDA

- Introduction to EU Projects
- PHYSICS EU Project
  - Consortium, WPs
  - Objectives
  - Components
  - Research topics and results
- Red Hat role and main technologies
- K8s Image Layer Scheduler: benefits of open source

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

INTRODUCTION TO EU PROJECTS

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

3 main types

- Research and Innovation Actions (RIA)
  - Typically aiming to deliver prototypes and integrated environments (TRL 3-4-5)
- Innovation Actions (IA)
  - Start from existing prototypes (TRL3-4) and increase the TRL or target initial commercial services
- CSA actions
  - Support actions aiming at standardization, organization of dissemination etc.
- Budget Sizes: depends on the EC call for proposals

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

EC calls

- Centered around 7-year Framework Programs
  - FP7 (2007-2013), H2020 (2014-2020), Horizon Europe (2021-2027)
- Work programmes of 1-2 years ahead issued by the EC
- Consortia of partners organize and target a specific call:
  - PHYSICS RIA call example: https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/ict-40-2020
  - Call has a specific high level scope: Smart cloud/edge continuum
  - Balance between academic and industrial partners (both large players and SMEs)
  - Balance between activities: R&D (if a RIA), piloting applications, innovation, dissemination, exploitation
- Work is split into Work Packages (WPs) and tasks inside WPs

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

The PHYSICS Project

oPTimIzed HYbrid Space-time servIce Continuum in faaS

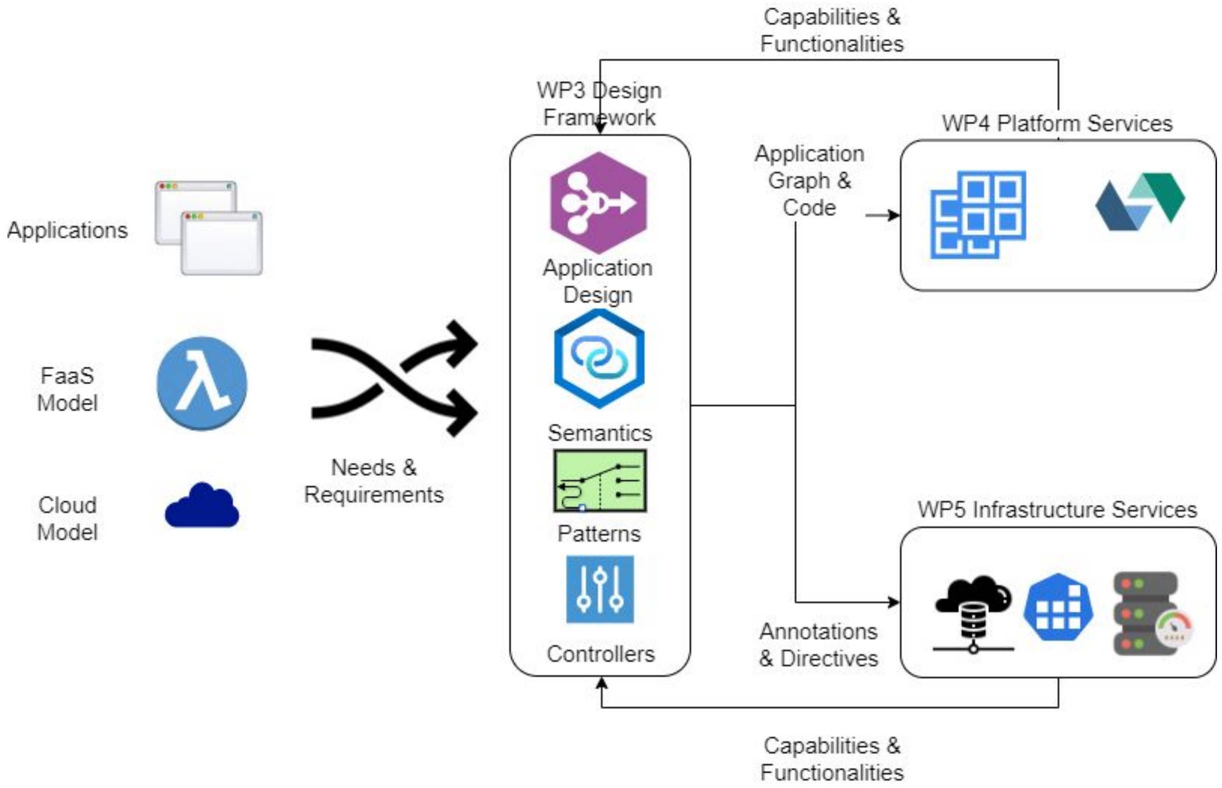
This project has received funding from the European Union's horizon 2020
research and innovation programme under grant agreement no 101017047

Challenges targeted by PHYSICS

- Abstract usage of **service offerings and clusters across the Continuum**
- Adaptation of code to new **serverless** computing paradigms
- Investigation of **space** (location of execution)-**time** (duration of execution) in the continuum
- Optimization of resource **selection** and operation (**global** and **local** level)
- Multiple Exploitation Channels and **Reusable** Artefacts

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Project Goals



Visual programming environment to create serverless workflows with reusable patterns and increased semantics

Platform-level functionalities to orchestrate and deploy FaaS workflows and optimize cloud/edge interplay

Provider-local resource management mechanisms to offer competitive and optimized services execution

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Project details

- H2020 ICT 40 - Cloud Computing: towards a smart cloud computing continuum
- Research & Innovation Action
- Project Budget: 4,985,712€
- EC Grant (100%): 4,985,712€
- Timeframe: 01-01-2021 to 31-12-2023
- 14 Partners
- GFT is Project Coordinator

	1. GFT – Italy		8. INQBIT – Romania
	2. ATOS IT – Spain		9. INNOVATION SPRINT – Belgium
	3. HPE – Italy		10. INNOV-ACTS – Cyprus
	4. RED HAT – Israel		11. CYBELETECH – France
	5. FUJITSU – Germany		12. UPM – Spain
	6. BYTE COMPUTER – Greece		13. HUA – Greece
	7. RYAX TECHNOLOGIES - France		14. DFKI - Germany

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

High Level Project Plan

M1-M7	M8-M18	M19-M23	M24-M34	M35-M36
PHASE 1	PHASE 2	PHASE 3	PHASE 4	PHASE 5

PHASE 1

DESIGN
- Market gap analysis
- Requirements gathering
- Overall architecture
- UC scenarios

PHASE 2

IMPLEMENTATION (1st ITERATION)
- Core features and implementation development
- Internal UC validation

PHASE 3

1st IMPACT
- Collect feedback from external audiences including EC

PHASE 4

IMPLEMENTATION (2nd ITERATION)
- final implementation round targeting the necessary extended functionalities

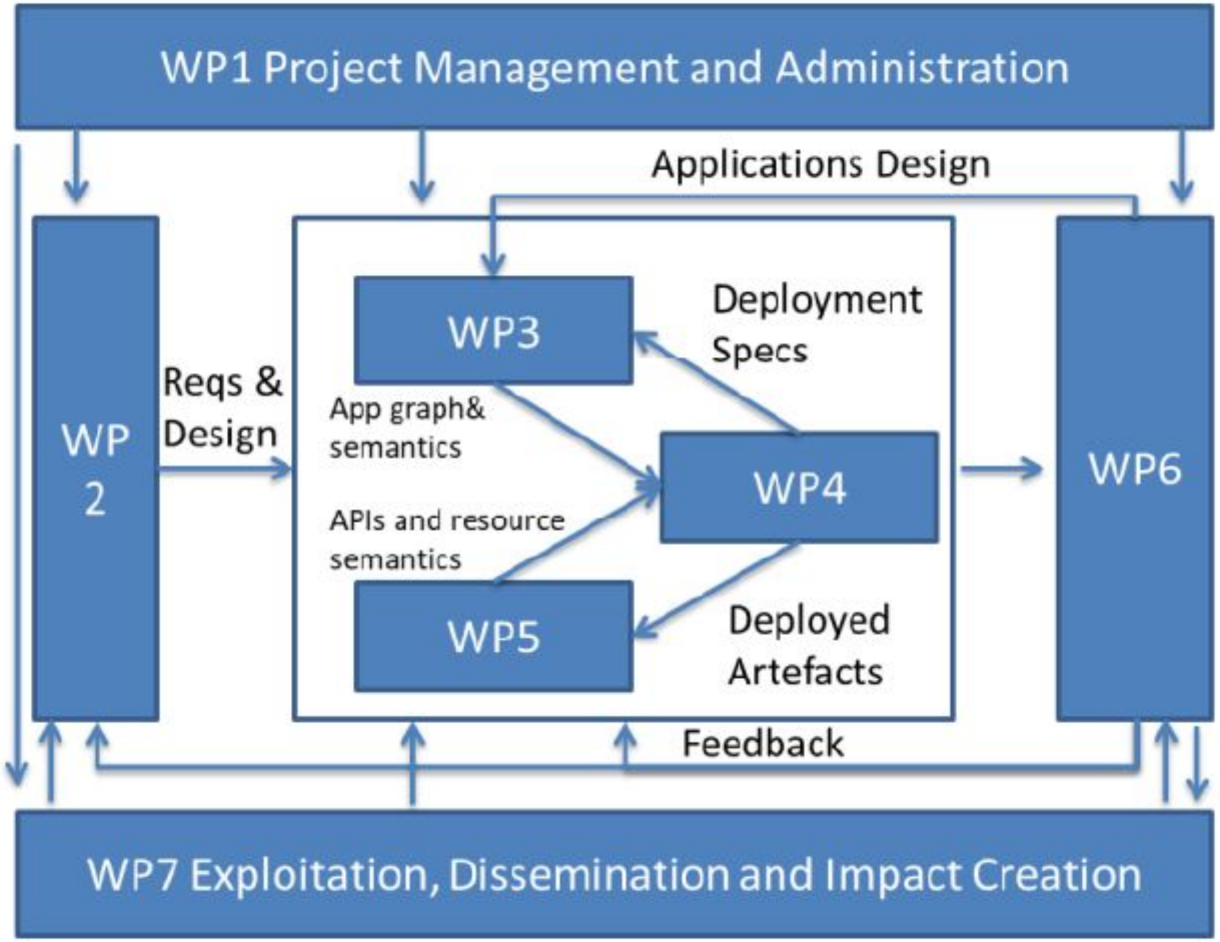
PHASE 5

2nd IMPACT
- Validation, usability, business evaluation and impact creation

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

WPs

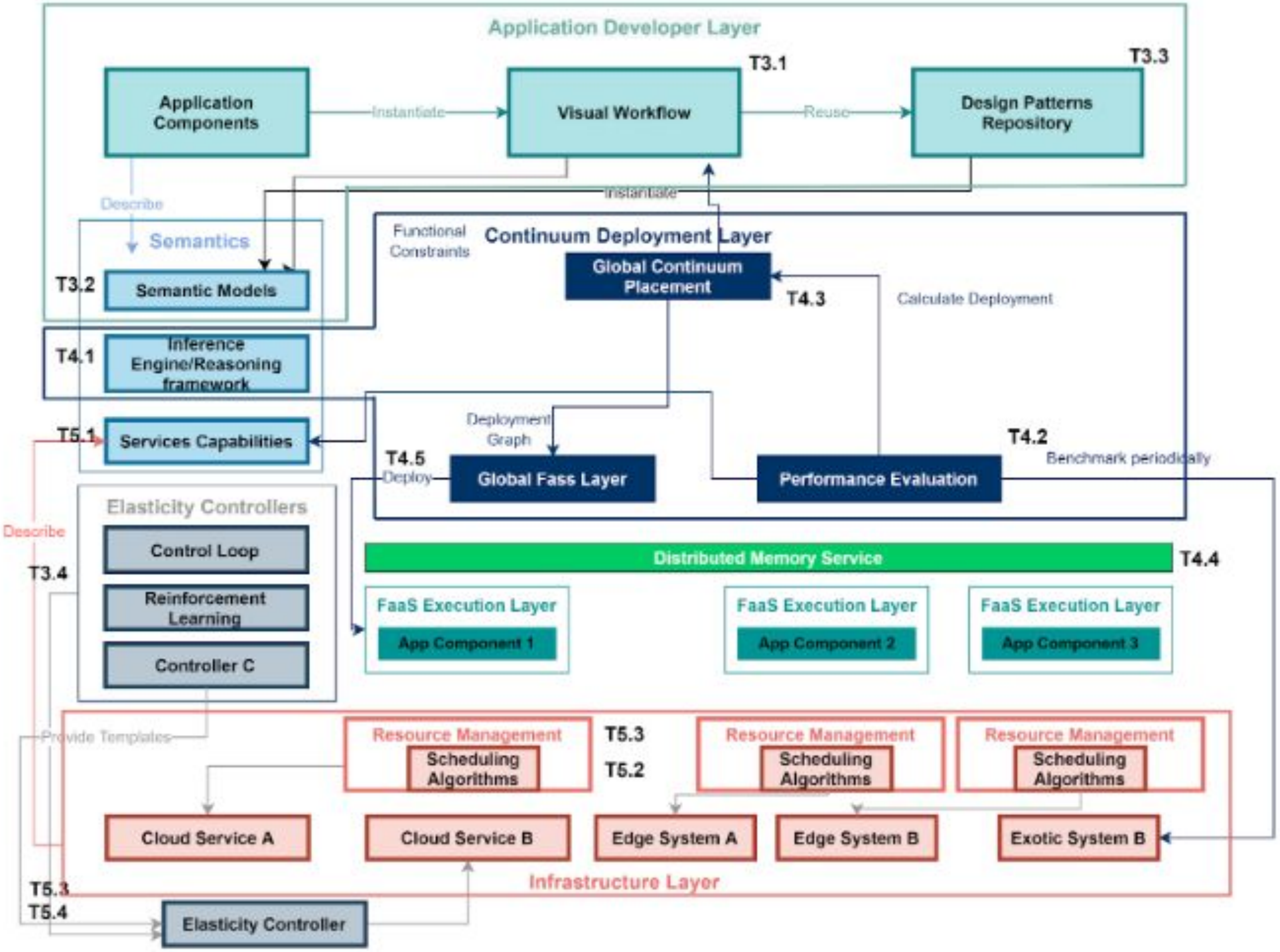
- WP2: Requirements and Architecture
- WP3: User Layer
  - Design Environment & Function DevOps
  - Reusable Patterns, semantics and handles from WP4/5
- WP4: Platform Services
  - Semantics Management
  - Placement Optimization
  - Performance Analysis
  - Data Services
  - Deployment Orchestration Services
- WP5: Infrastructure Services
  - Resource Controllers & Semantics
  - Local placement optimizers (container node selection)
- WP6: Applications, Pilots and Platform Integration



This project has receiv[...] research and innovation programme under grant agreement no 101017047

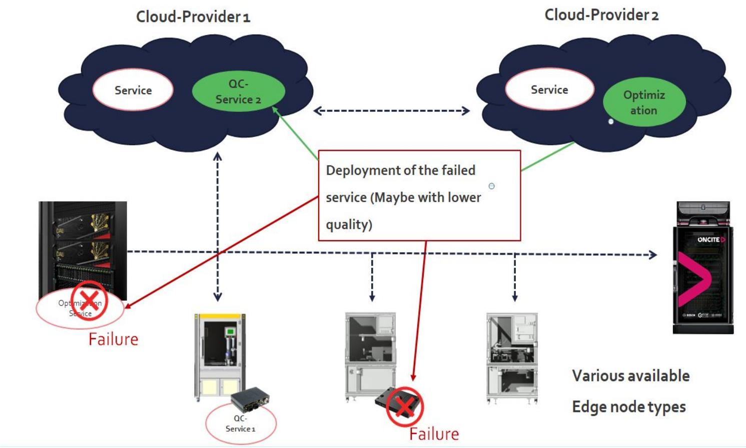
PHYSICS Technical Architecture

- WP3: User Layer
  - Design Environment & Function DevOps
  - Reusable Patterns, semantics and handles (e.g. controllers) from WP4/5
- WP4: Platform Services
  - Semantics Management
  - Placement Optimization
  - Performance Analysis
  - Data Services
  - Deployment Orchestration Services
- WP5: Infrastructure Services
  - Resource Controllers & Semantics
  - Local placement optimizers (container node selection)

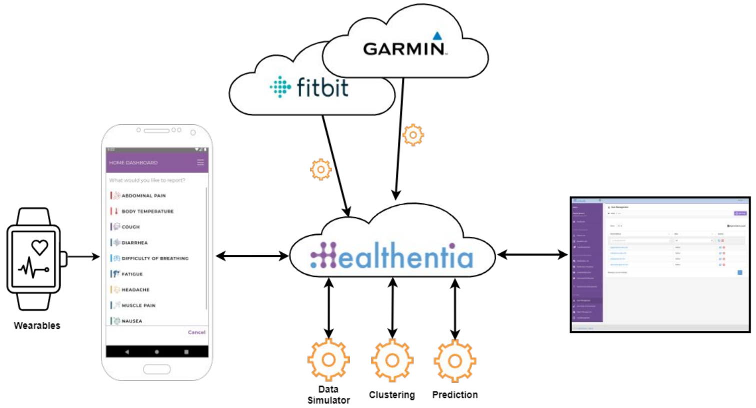


Project Pilots

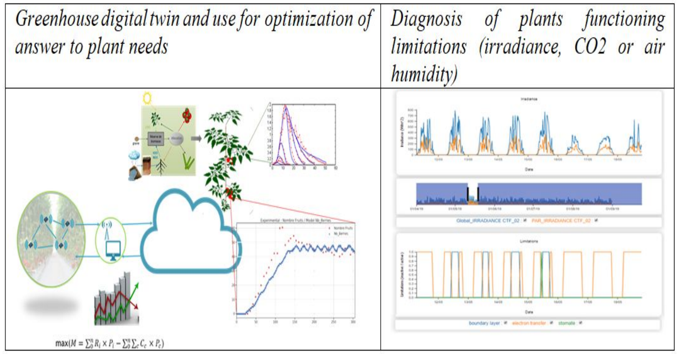
Smart Manufacturing for increased resilience and interplay



eHealth Personalized Monitoring and Collective Analysis



Smart Precision Agriculture



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

PHYSICS

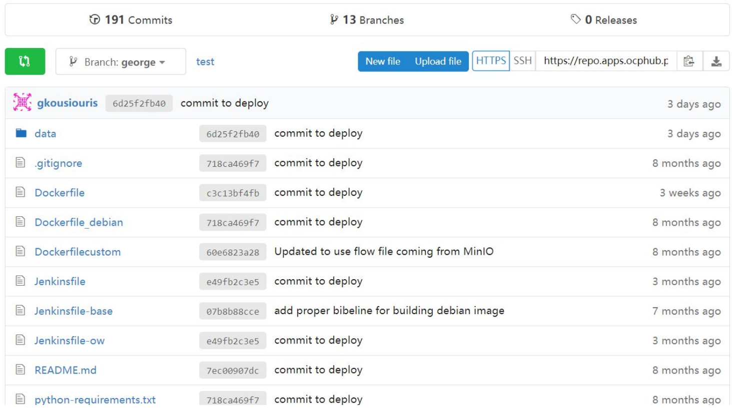
PHYSICS Design Environment

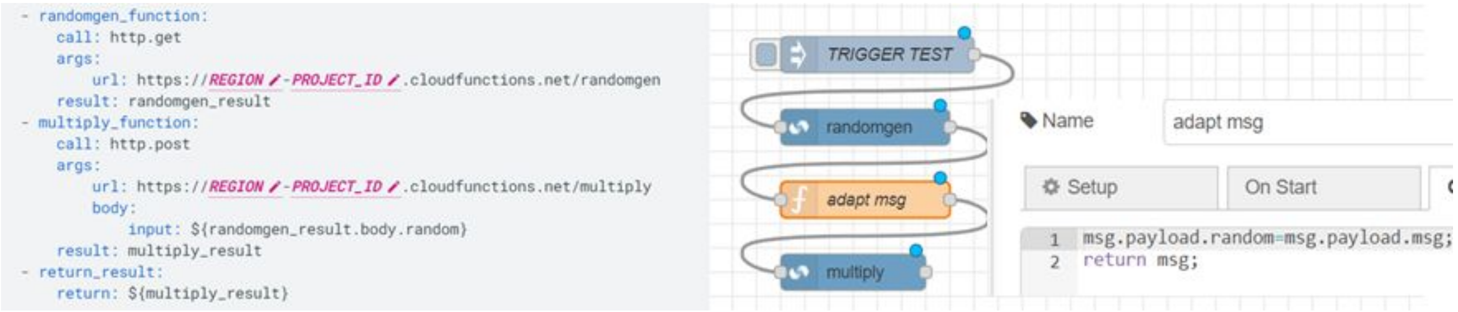
Joint Work from GFT, HPE and HUA

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

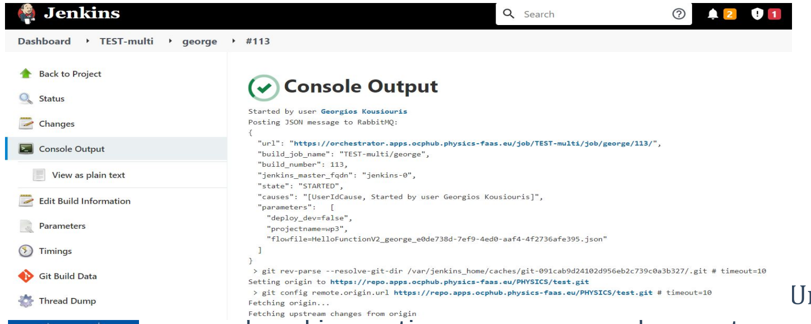
Goals of PHYSICS DE: Enhance, Abstract, Enrich FaaS application creation

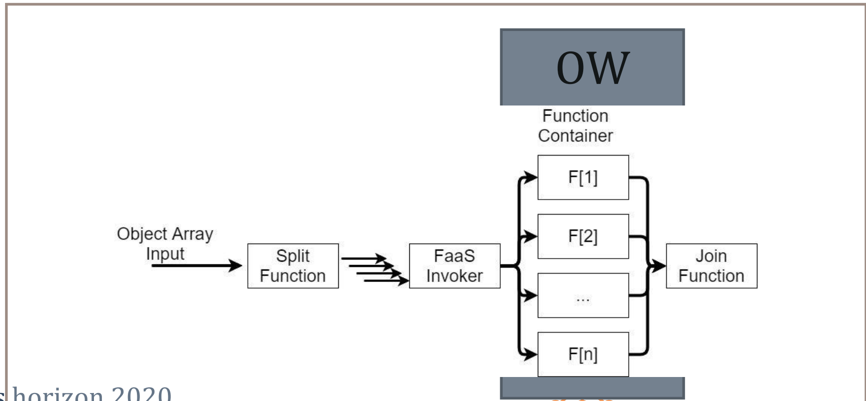
- Customize function environment
- Simplify the creation of complex function workflows
  - Support complex primitives like Joins
- Exploit reusable patterns
- Increase semantics
- Abstract packaging, testing and deployment





Case	Measured Quantity
Number of errors	10
Errors solved with Node-RED test	7
Errors needing FaaS deployment	3
Local Node-RED test	1-2 seconds
FaaS deployment and test (including image creation)	5 minutes
Time saved	35 minutes

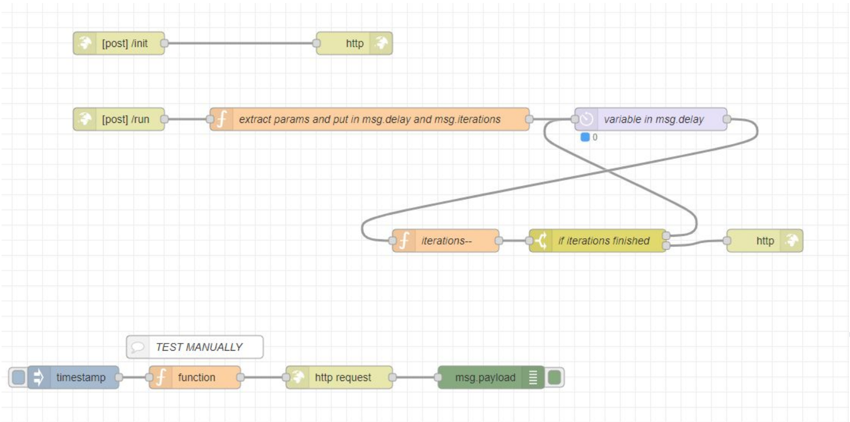


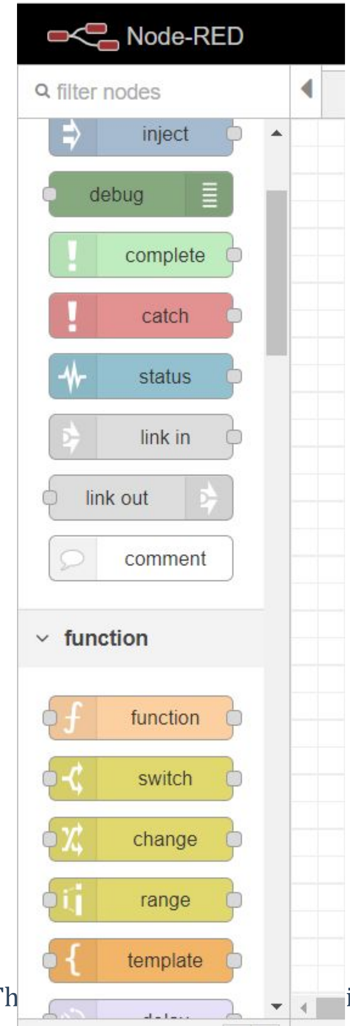


Union's horizon 2020 research and innovation programme under grant agreement no 101017047

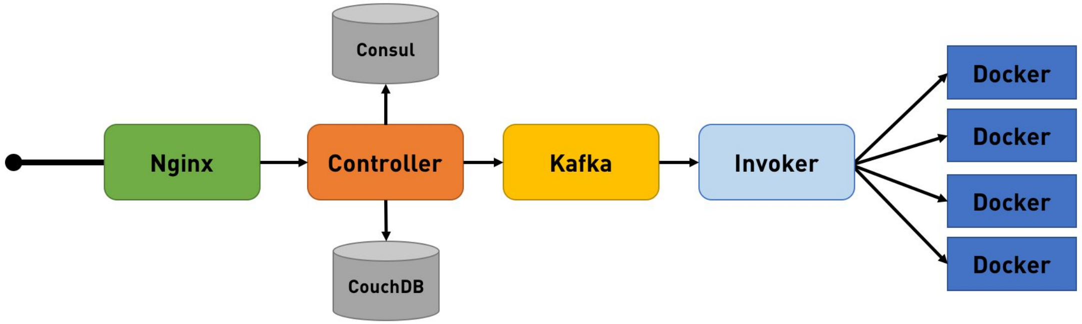
Baseline Technologies

- Node-RED
  - Programming environment for event driven applications
    - Built-in nodes
    - NPM extension nodes
    - Subflows (groups of functions) as nodes
  - Combined workflow orchestration and function execution abilities
  - Used as the main execution **runtime** and function **choreographer** for PHYSICS functions



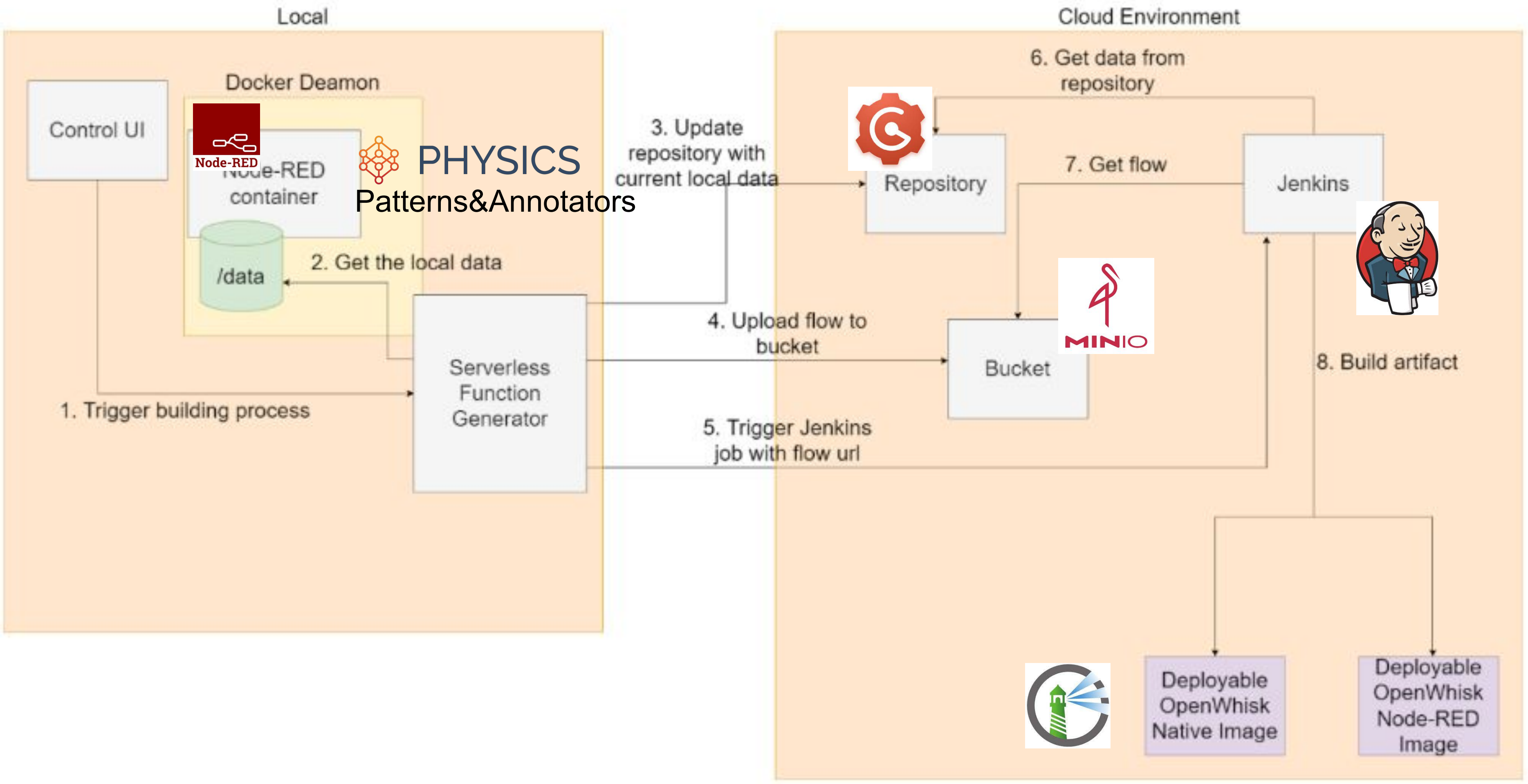


- Openwhisk
  - Open source FaaS platform
  - Supported by IBM
  - Main tool behind IBM Cloud Functions Service
  - Can execute functions based on custom container images



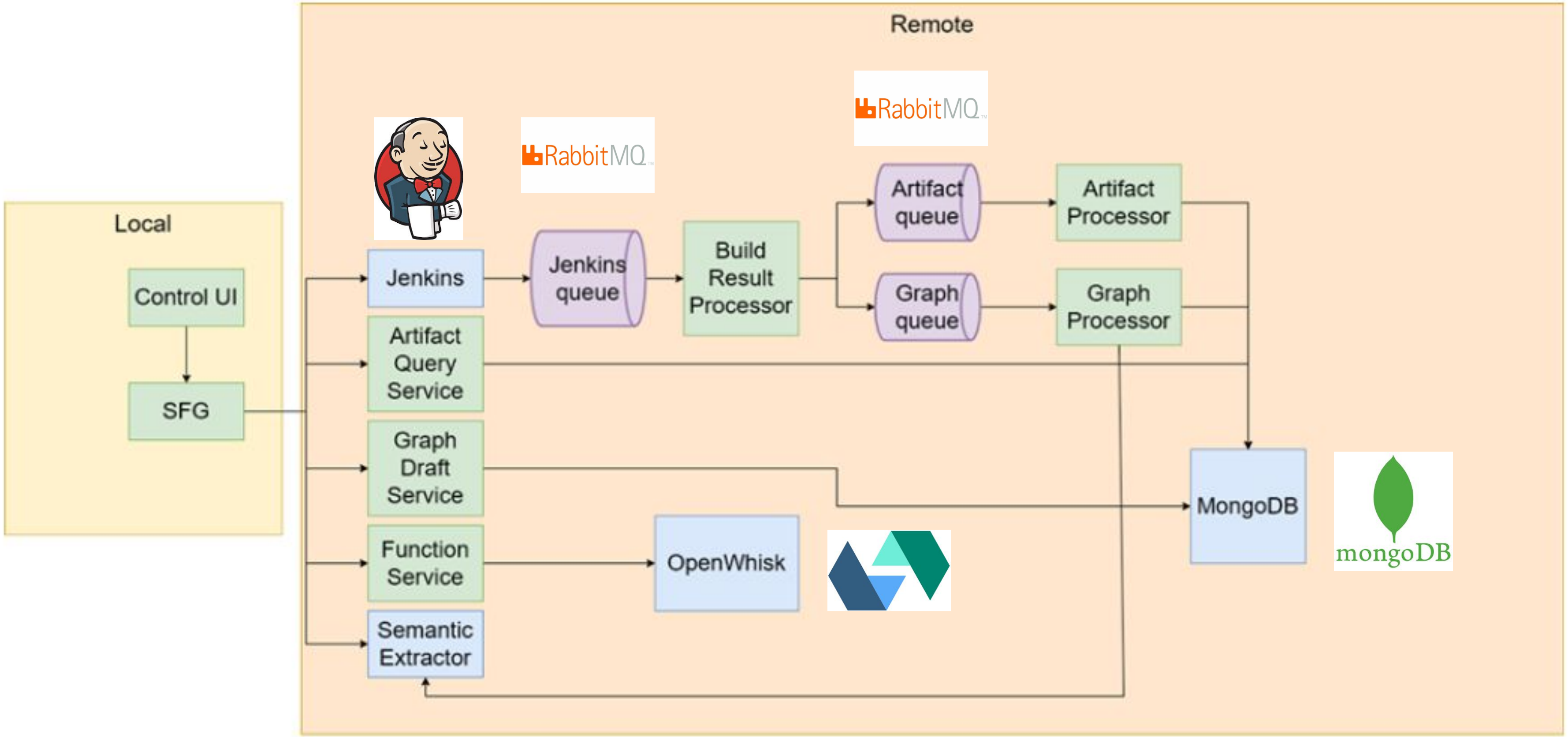
Th… ing from the European Union's horizon 2020
res… programme under grant agreement no 101017047

PHYSICS Design Environment Function Artefact Creation Lifecycle



research and innovation programme under grant agreement no 101017047

PHYSICS Design Environment Backend

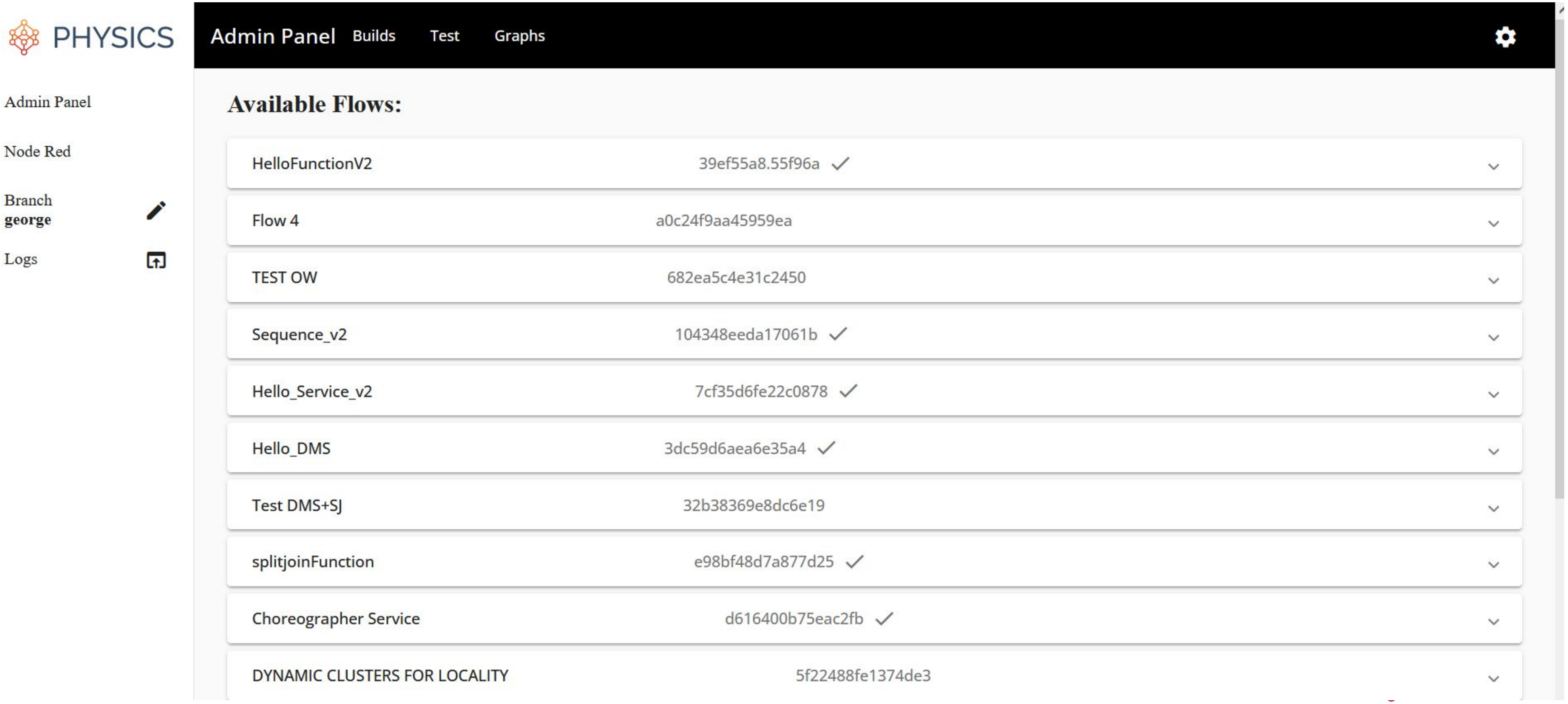


This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

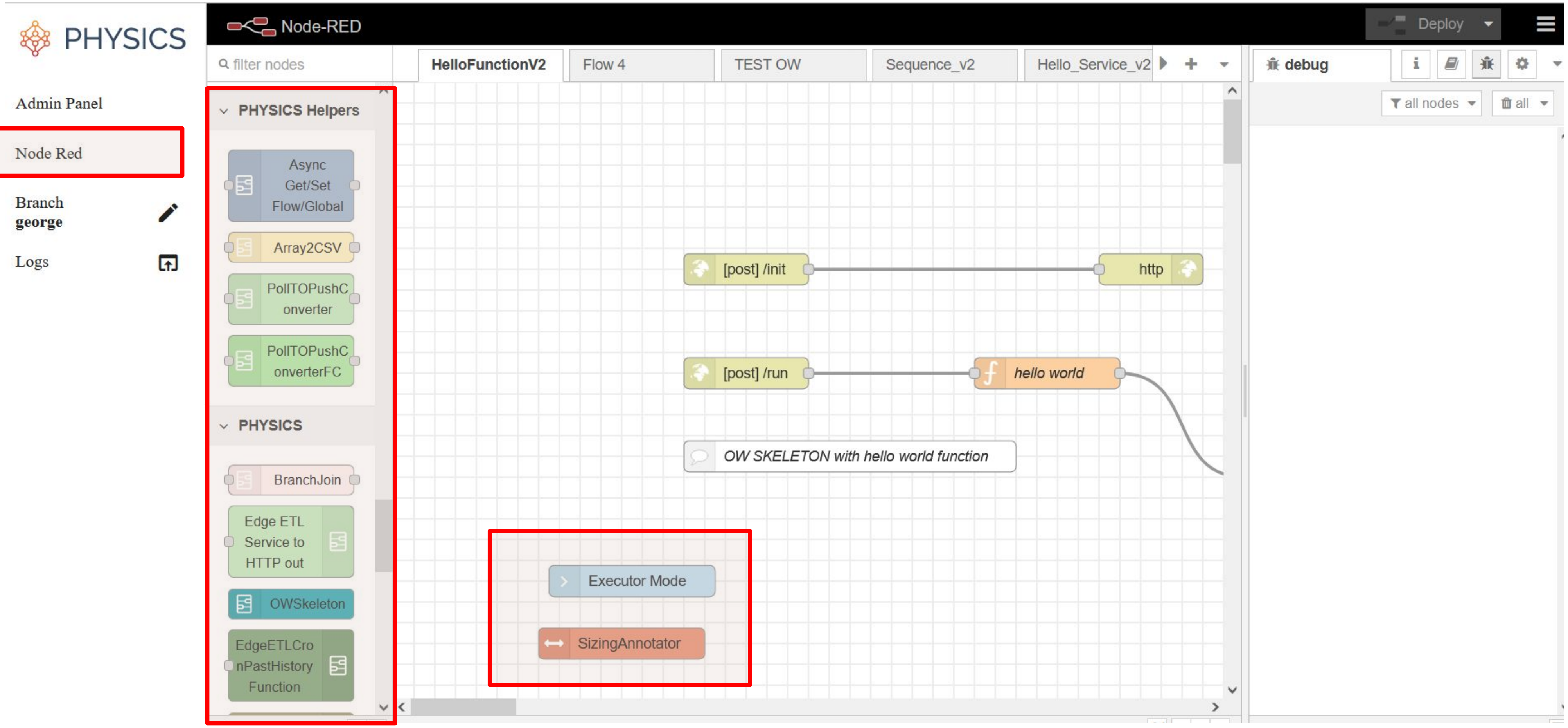
Design Environment Demo

- Full video available at: https://www.youtube.com/watch?v=DO2sFdfCD-o

- Function creation
- Function Build Managem ent
- Function Testing
- Logging
- App creation

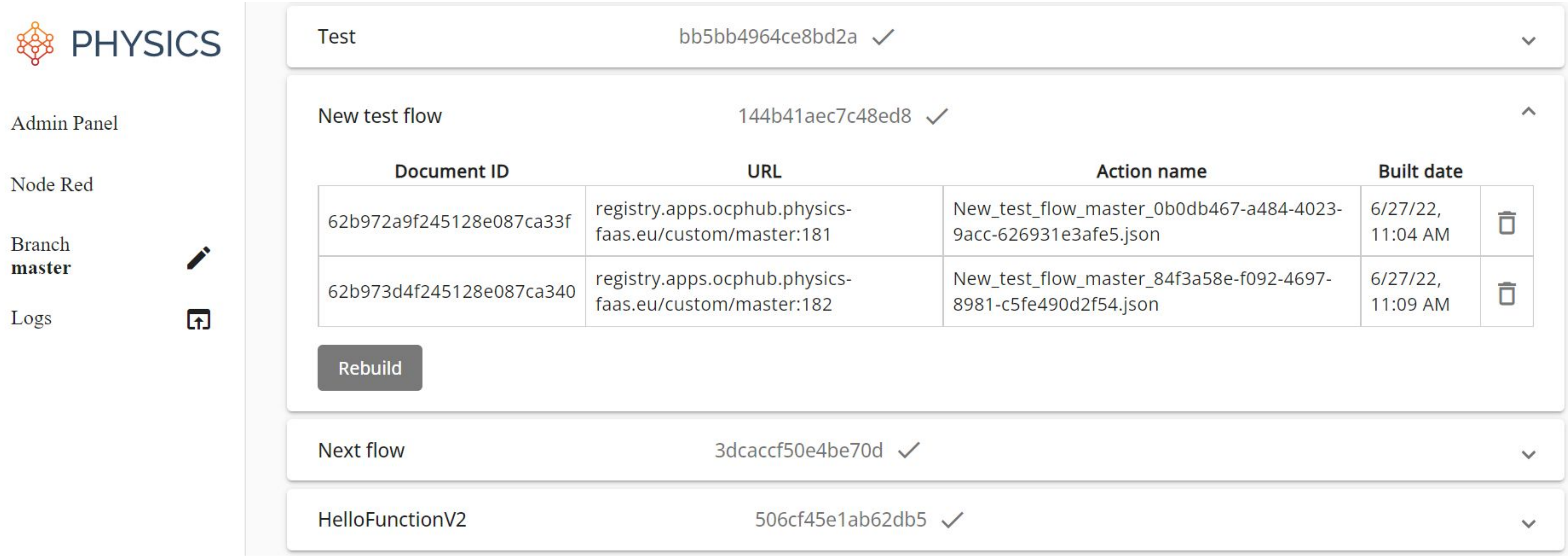


Embedded Node Red with enriched PHYSICS patterns and annotators- used to create and annotate flows



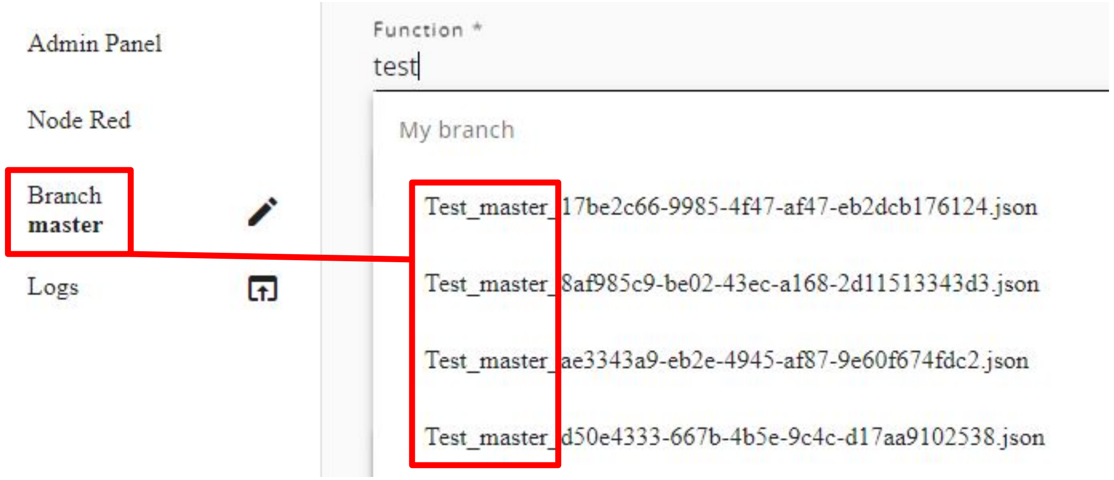
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

View and build list of available functions, created in Node Red tool.

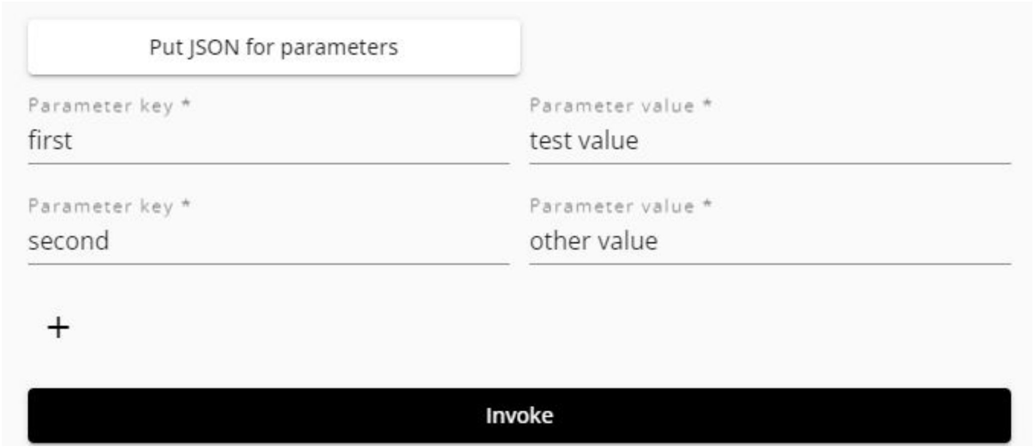


This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047

Invoke OpenWhisk functions built by user or for other branches

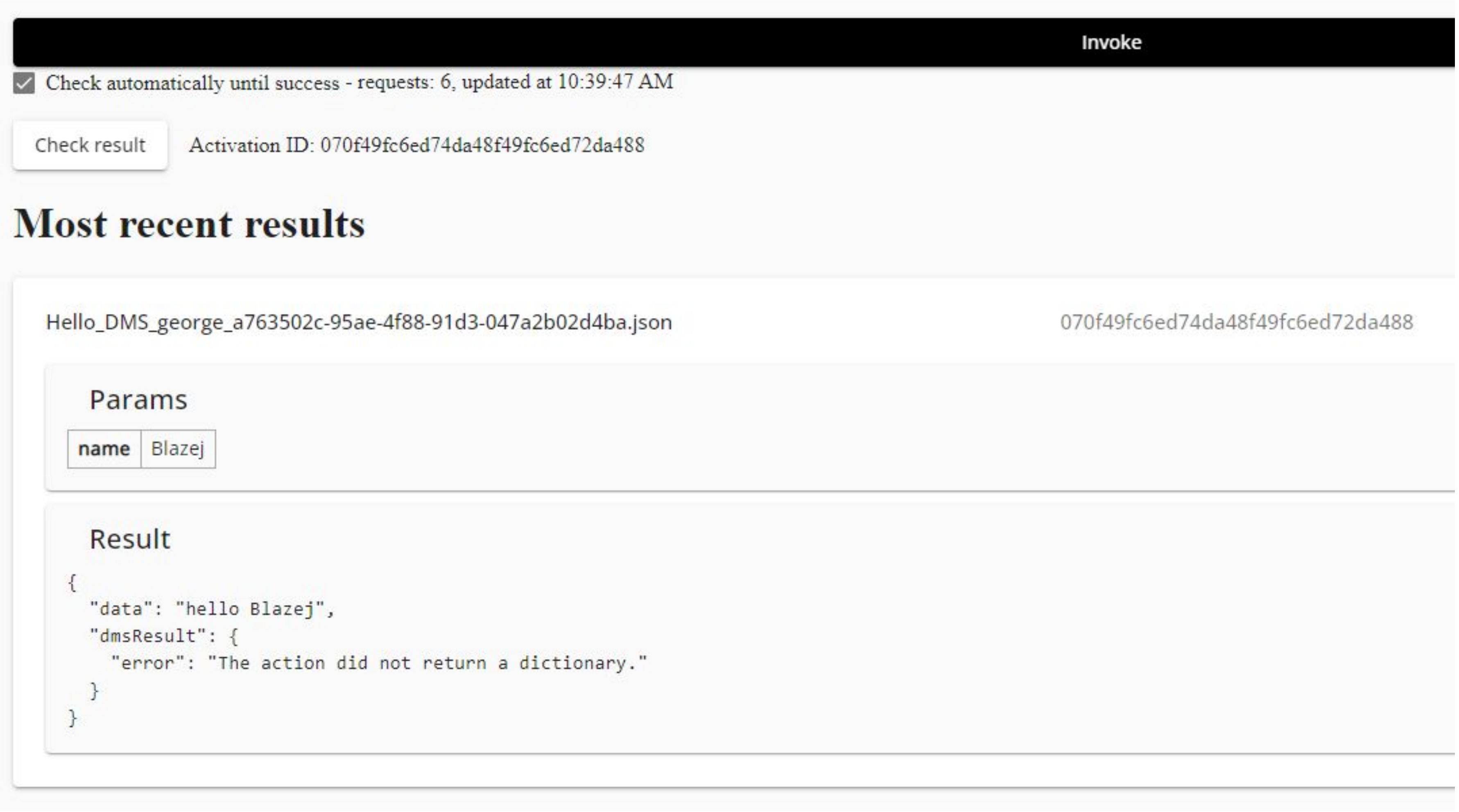


Setting parameters for invoke



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Example view for the result of function invocation.



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

App creation from selected graphs: package and manage multiple functions as one application for deployment

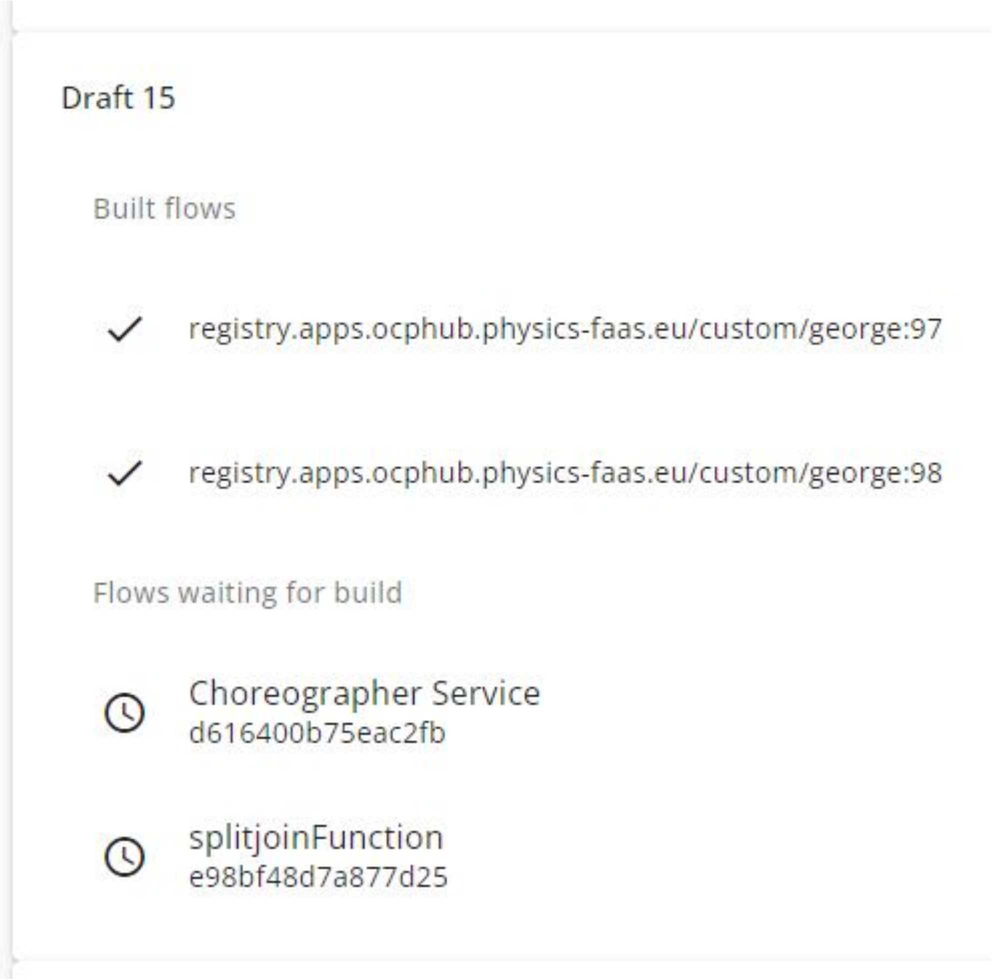
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Created Draft view

Consists of the flows that are already built

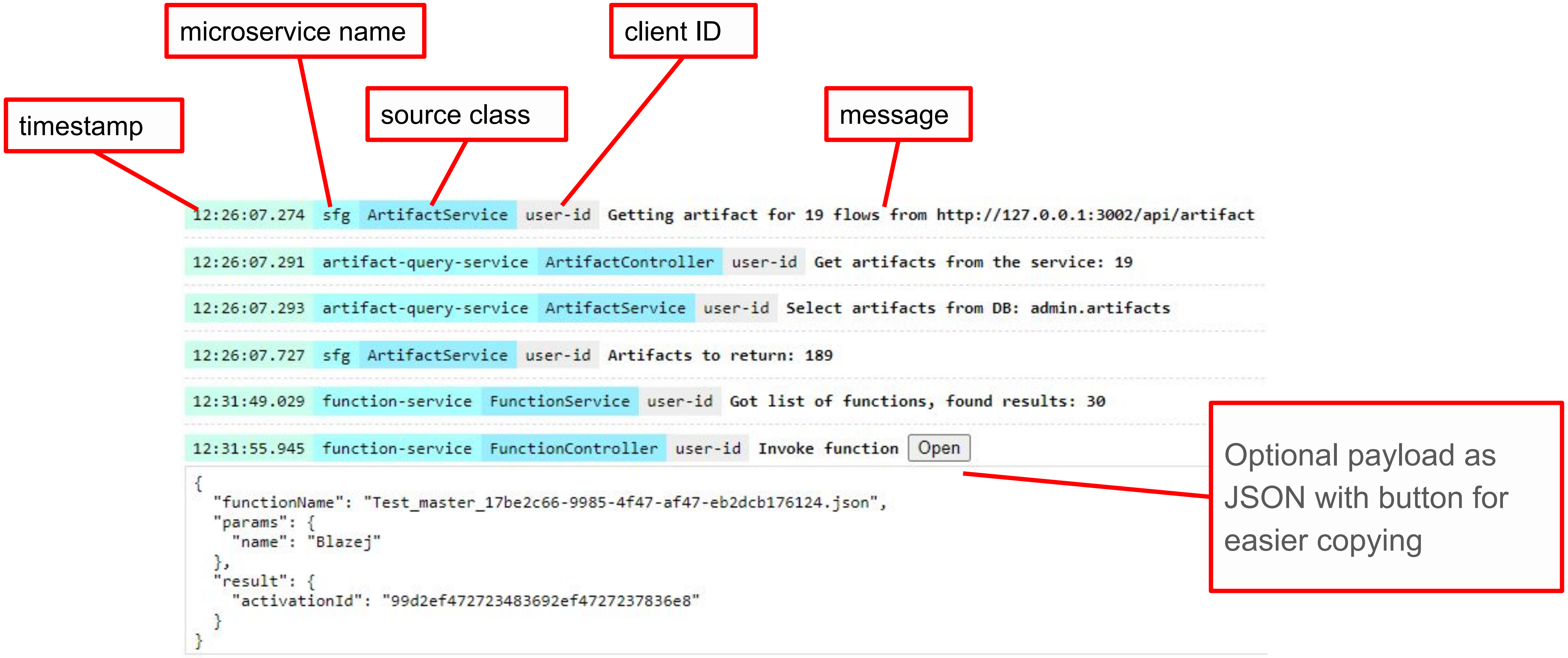
and

Flows for which had been triggered built process.



This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047

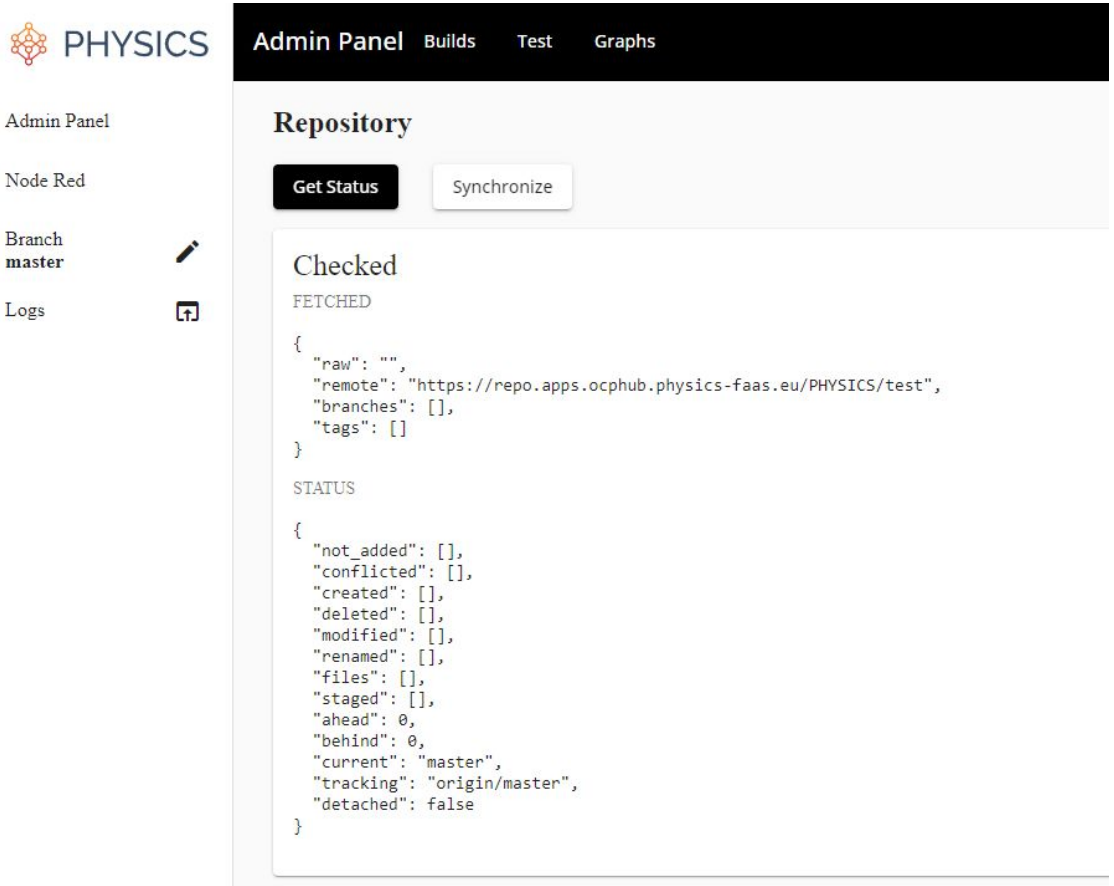
Displaying logs from microservices with useful information



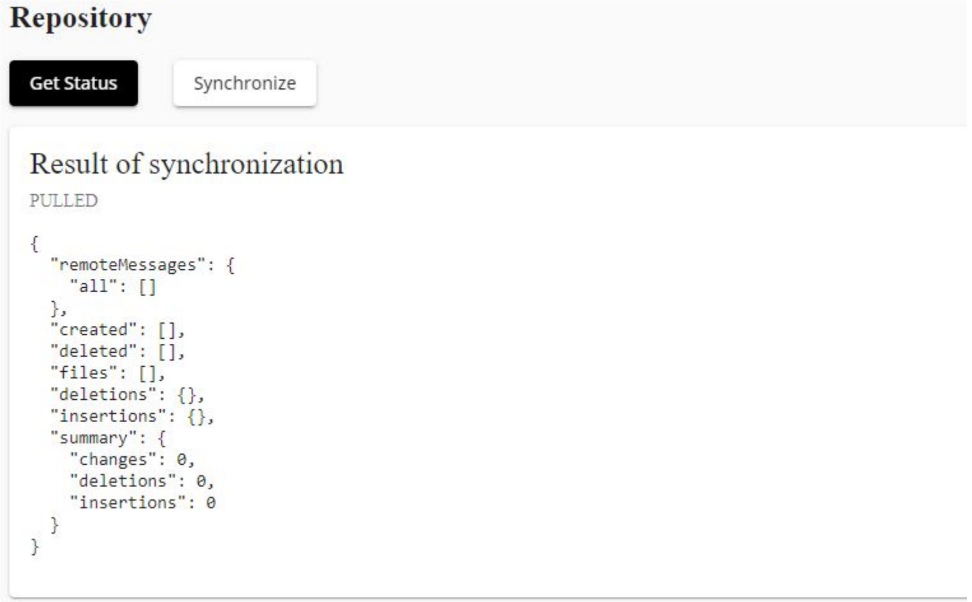
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

User repository synchronization tool

View for checking the status

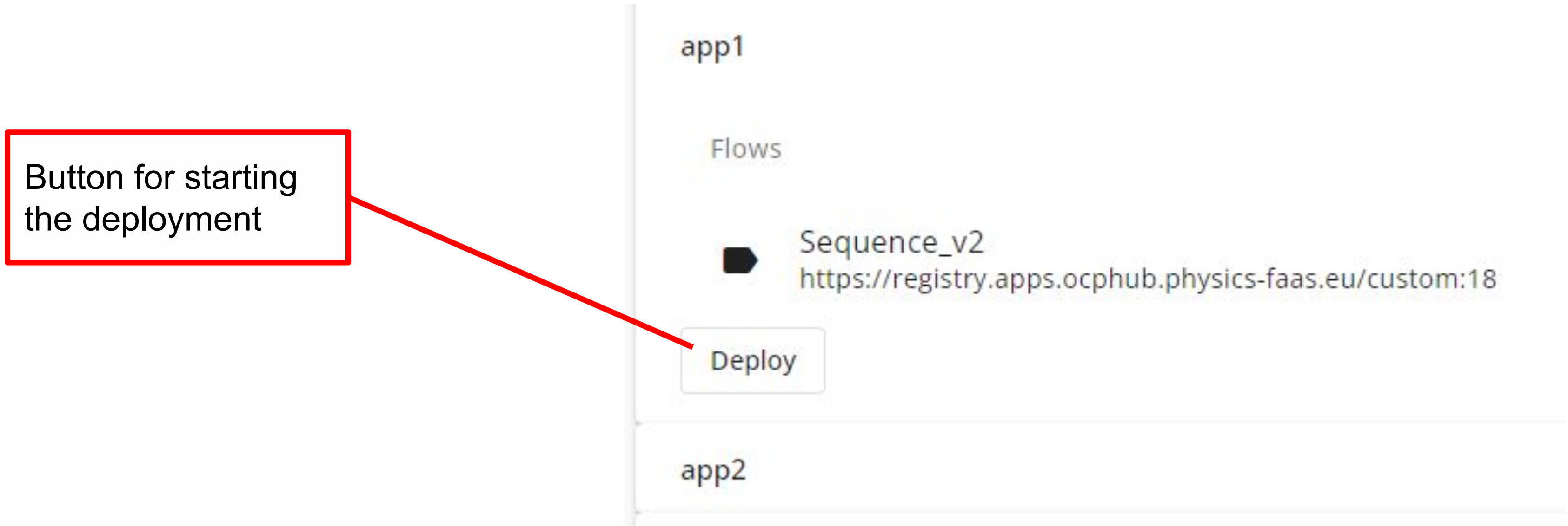


Result of the synchronization of the repository



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

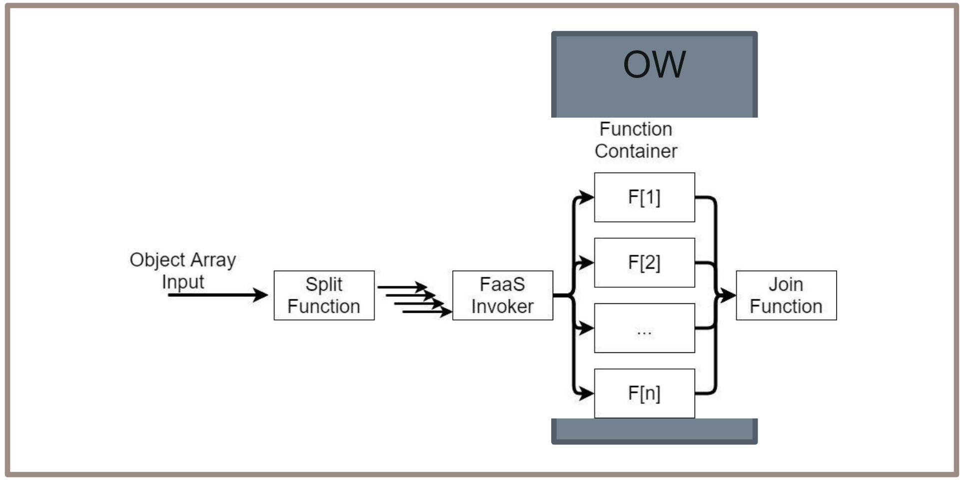
Deployment status

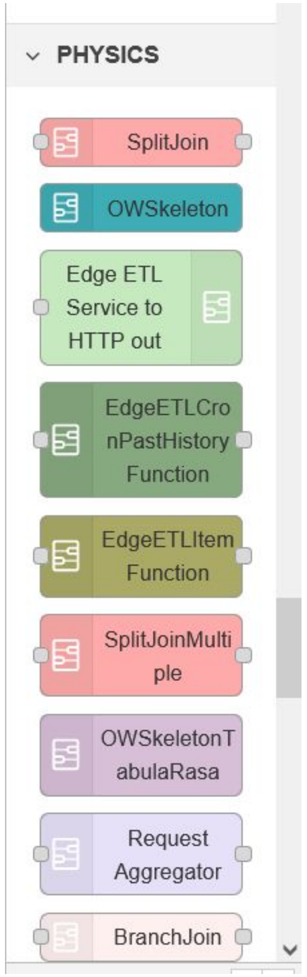


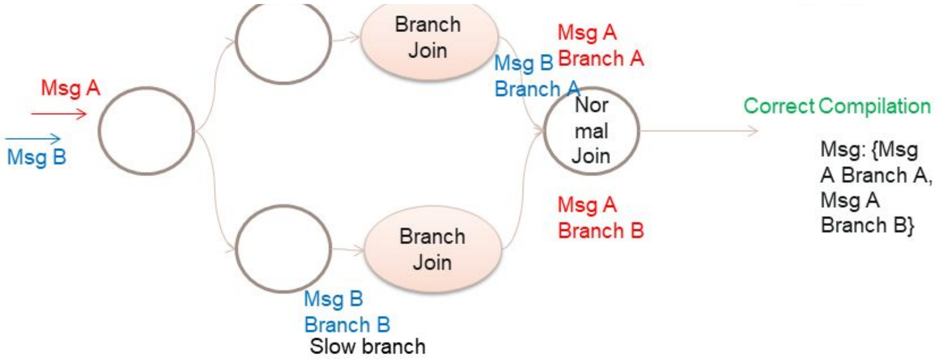
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

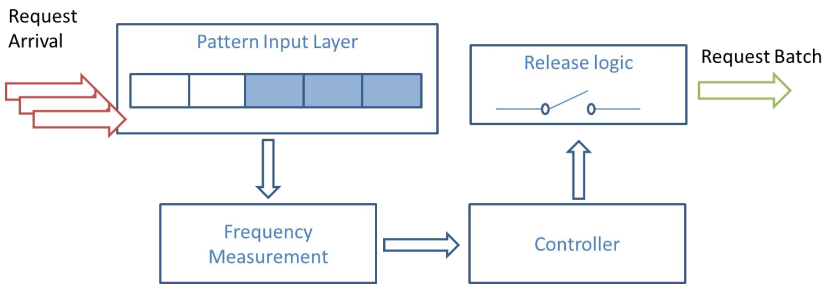
Patterns to support reusability, manageability and abstracted functionality

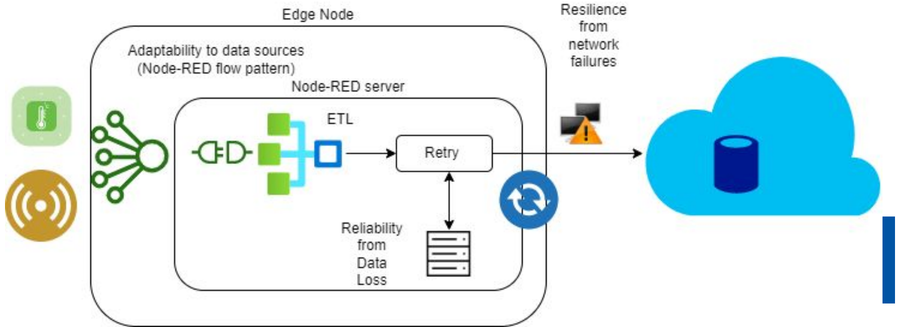
- Parallelization
- Context Management
- Data Collection
- Request Management
- Workflow primitives
- Etc.

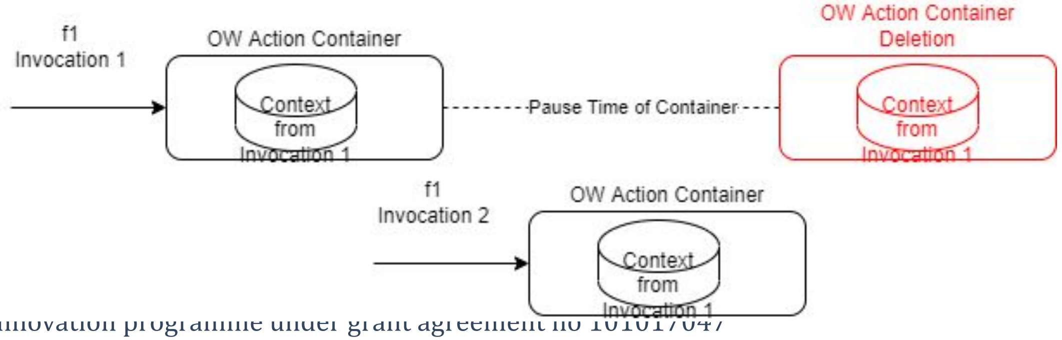






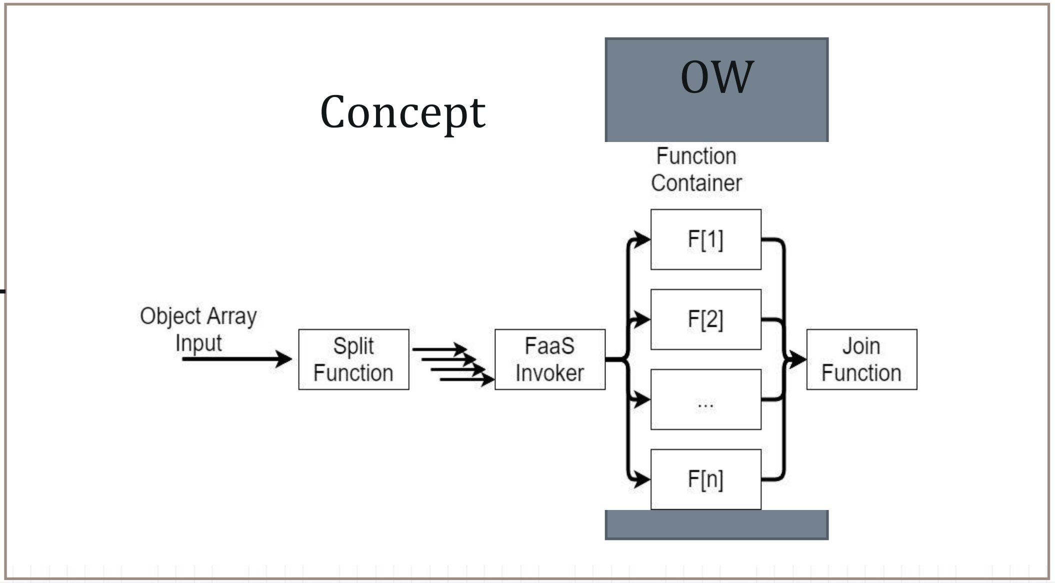




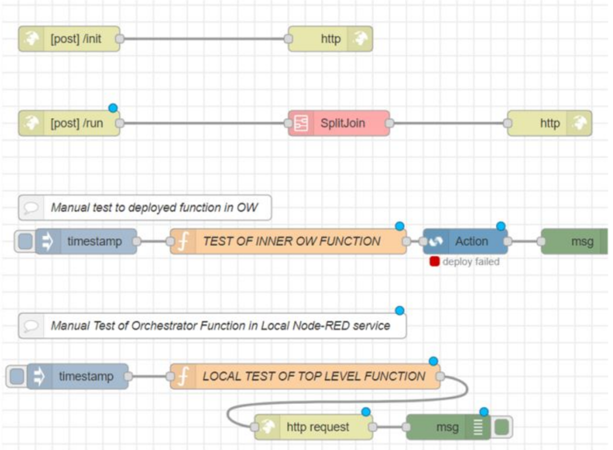


This project ha
research and innovation programme under grant agreement no 101017047

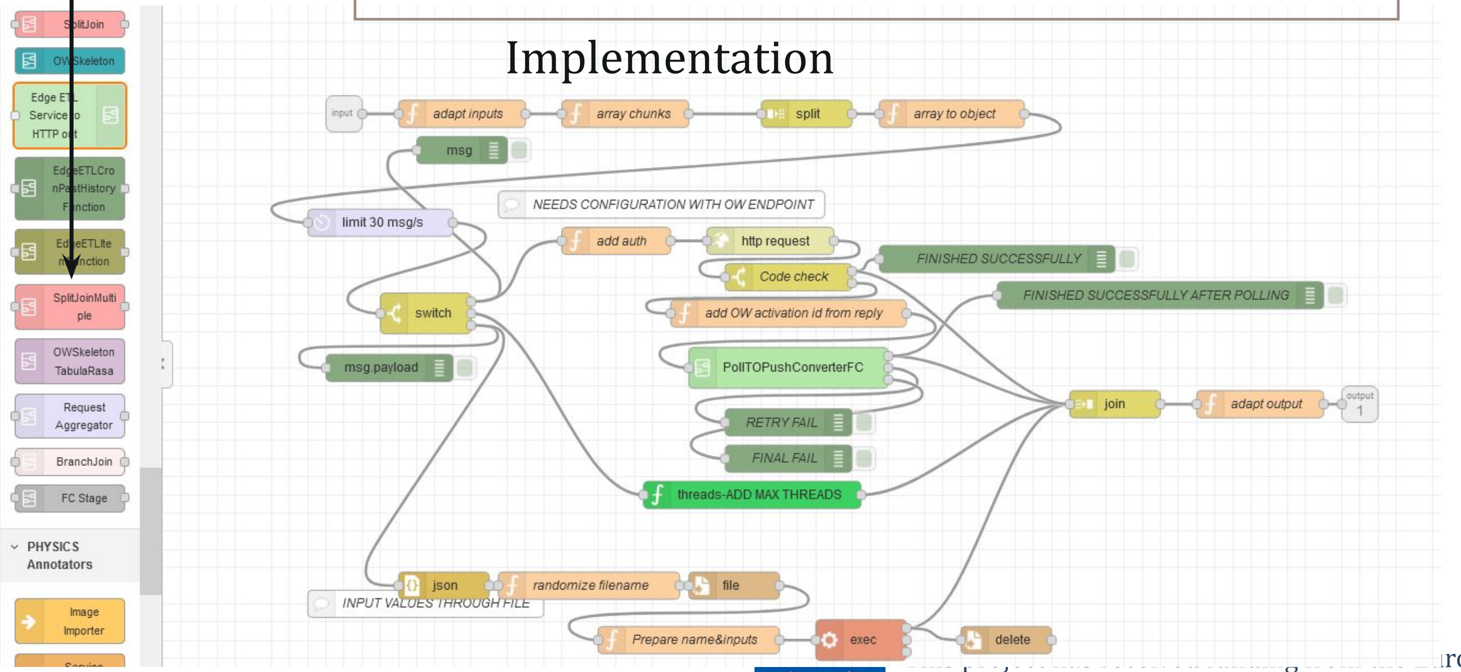
Example of Fork-Join parallelization pattern



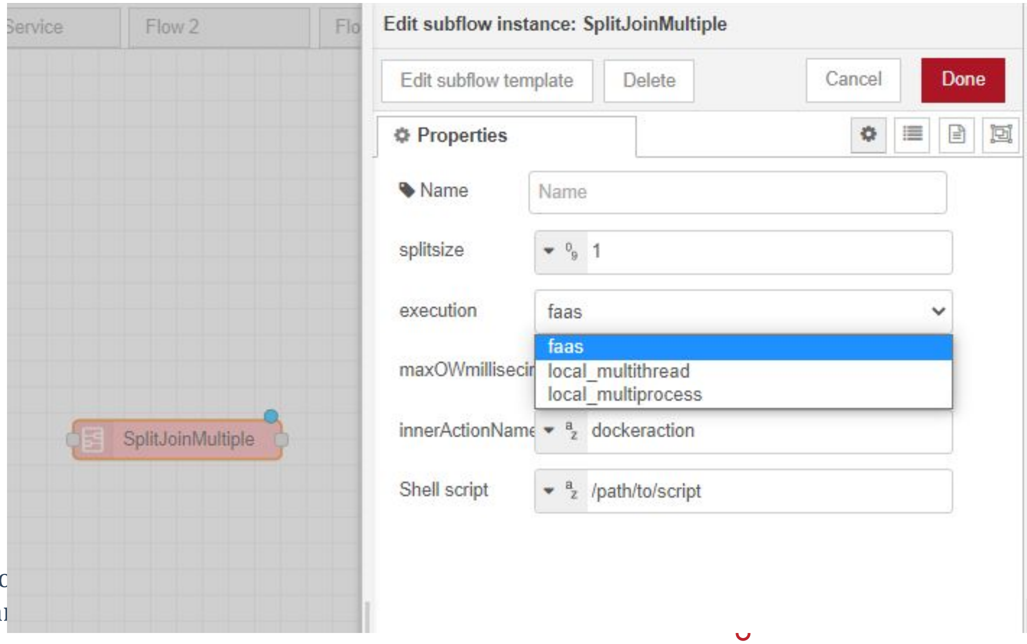
Usage in a Function



Implementation



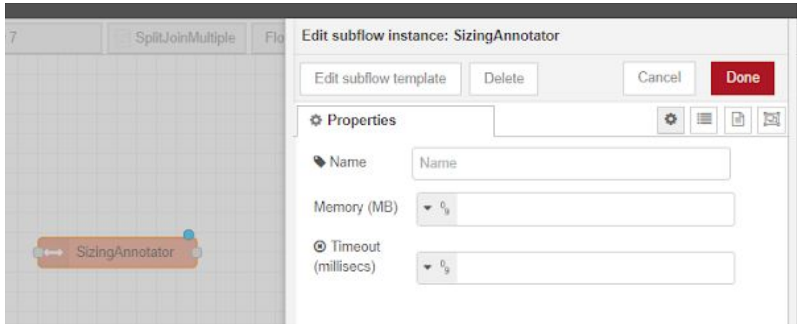
Parameterization

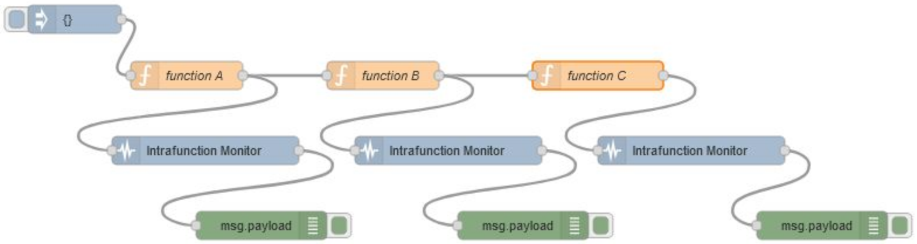


research and innovation programme under gra

Available semantic nodes

- Function relative placement (Affinity, Antiaffinity)
- Function Locality (e.g. Cloud/Edge)
  - Dynamically retrieves available clusters
- Importance (High/Medium/Low)
- OptimizationGoal (Performance, Energy etc)
- Function Sizing
- CPU Architecture requirements
- Intrafunction monitoring
- DMS interface for linking to data services
  - Can indicate data locality






```
"consumesREST":[ ⊞ ],
"definesInterface":[ ⊞ ],
"dms_job_id":"ss",
"dms_location":"https://openwhisk.apps.ocphub.physics-faas.eu/api/v1",
"dms_namespace":"wp4namespace",
"dms_operation":"dms-w",
"dms_pocket_dst_file":"dd",
"dms_size":"10",
"executorMode":"NoderedFunction",
```

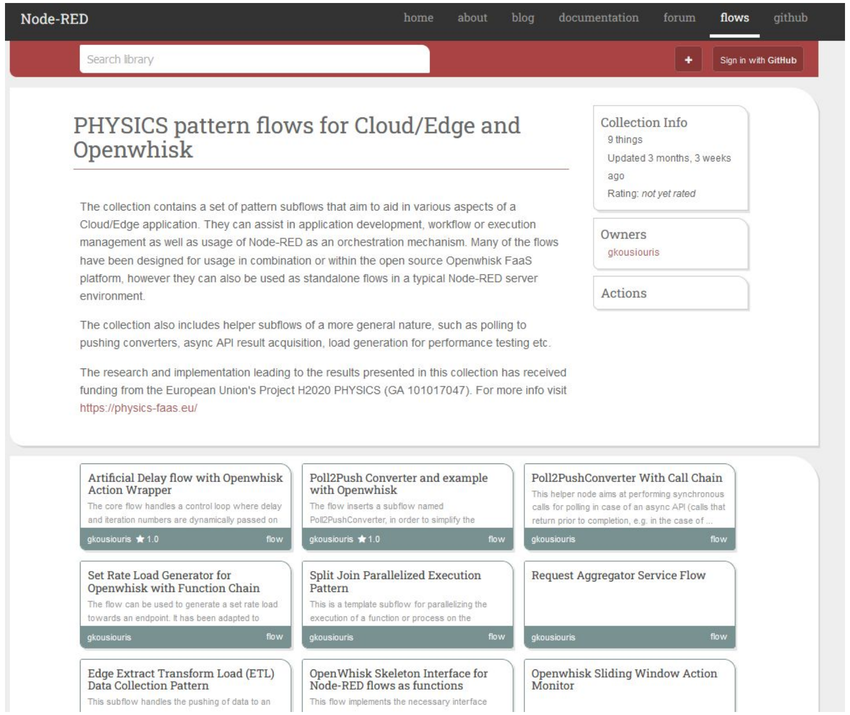

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

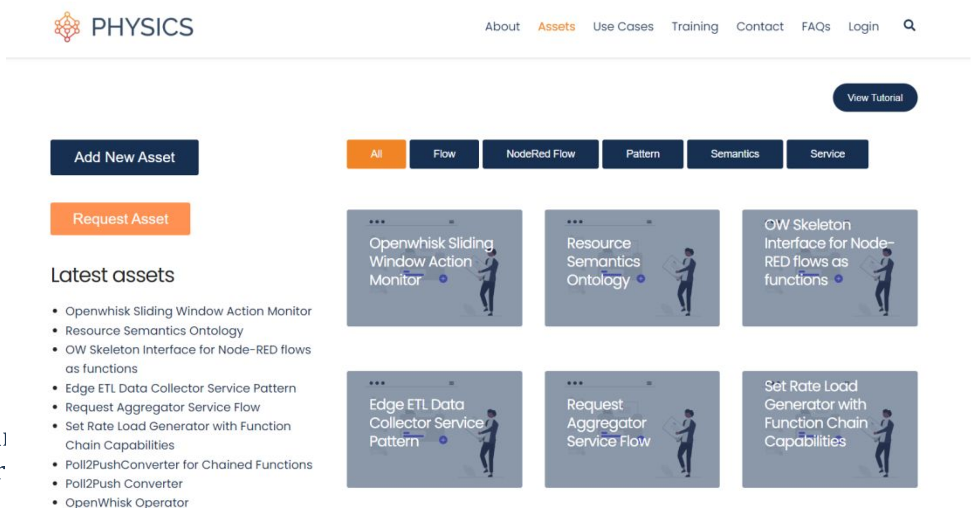
PHYSICS

oPtimized HYbrid Space-time servIce Continuum in faaS

Node-RED specific artefacts

- ❖ Pattern implementations as Node-RED reusable subflows
  - ❖ Exploitable as individual artefacts
  - ❖ Inclusion in Node-RED repo with documentation
    - ❖ https://flows.nodered.org/collection/HXSkA2JJLcGA
  - ❖ Included in PHYSICS RAMP as artefacts along with other PHYSICS components
    - ❖ https://marketplace.physics-faas.eu/assets

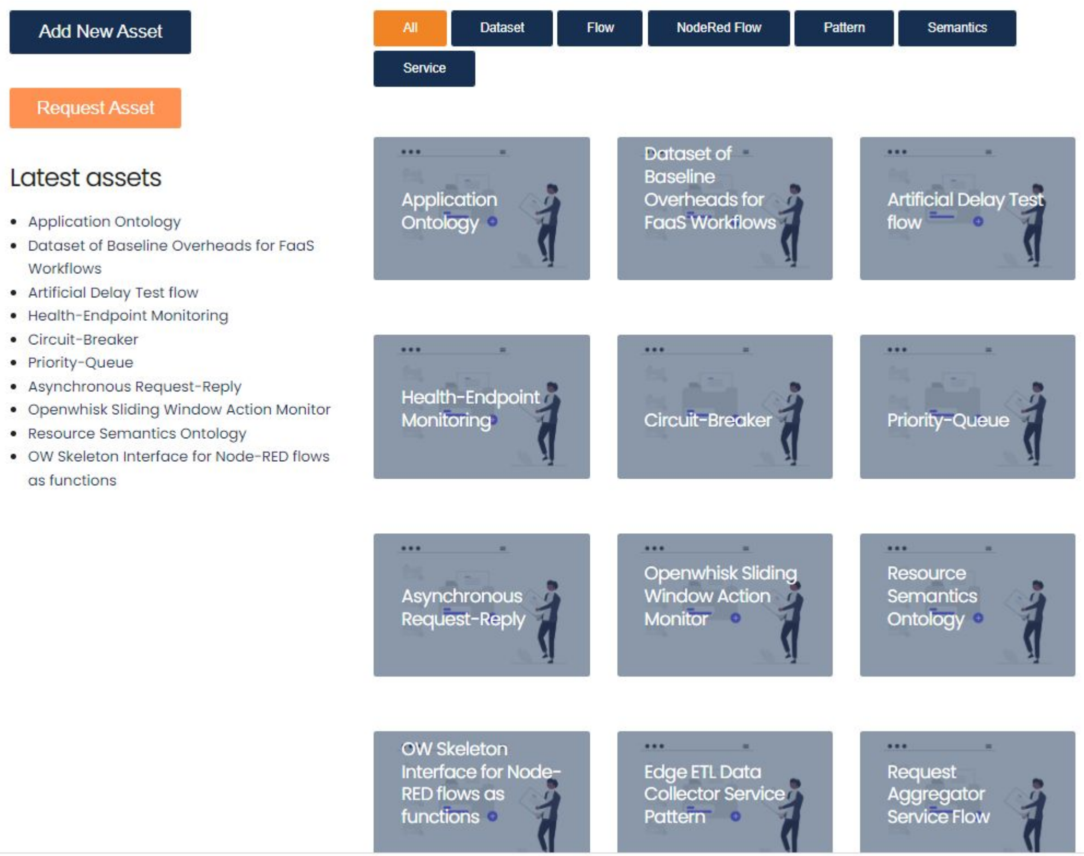




This project has received funding from the Eu
research and innovation programme under gr

PHYSICS RAMP

- Reusable Artefacts Marketplace Platform
  - https://marketplace.physics-faas.eu/assets
- Catalogue of available PHYSICS artefacts
  - Operators, Node-RED flows, Datasets, Services, Semantic Models
  - Links to repositories and usage guides
  - Ability to download and use
    - Looking for testers from the community!
  - Ability to request a specific asset or an extension to an asset



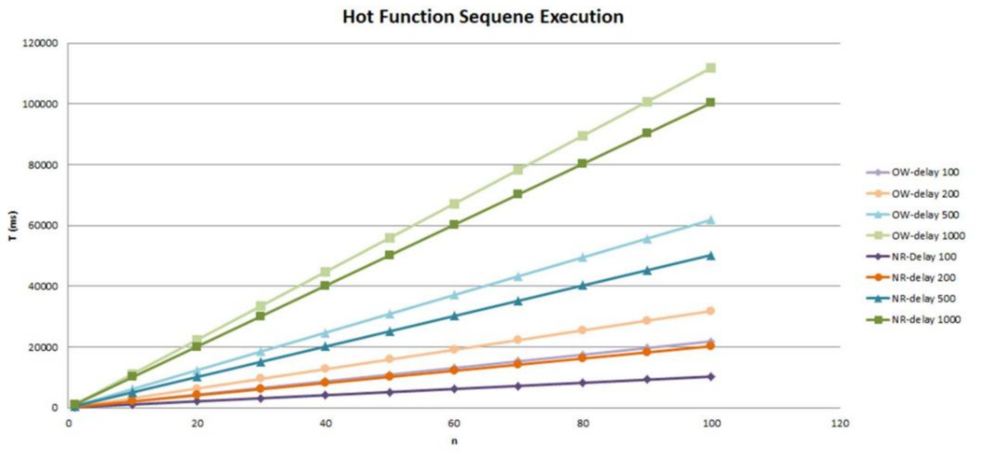
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

PHYSICS

PHYSICS Performance Aspects

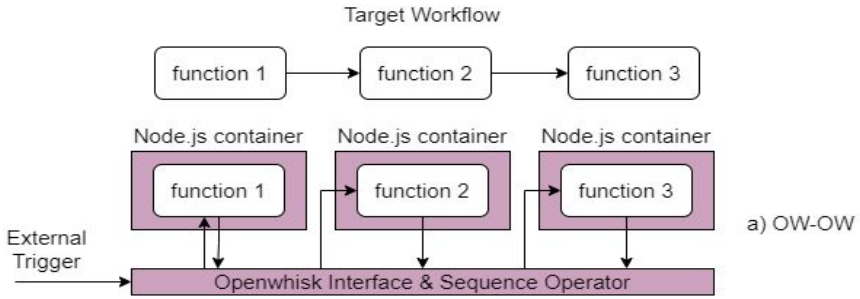
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Function Choreography Overheads in Different Modes

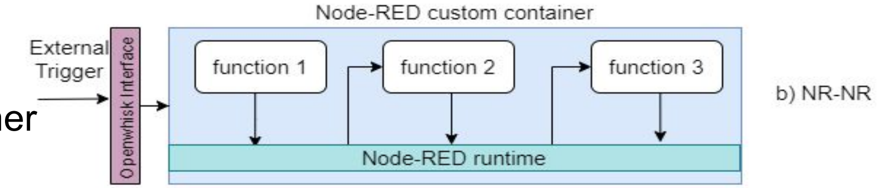


Mode	Warm Function Sequence Execution (ms)	Cold (ms)	Penalty
OW-OW	$T_{total}=2.11+n\times(T_{functiondelay}+118.94)$	+2248	
NR-NR	$T_{total}=128.36+n\times(T_{functiondelay} +1.69)$	+7373	
OW-NR	$T_{total}=114.08+n\times(T_{functiondelay} + 121.70)$	+9621	

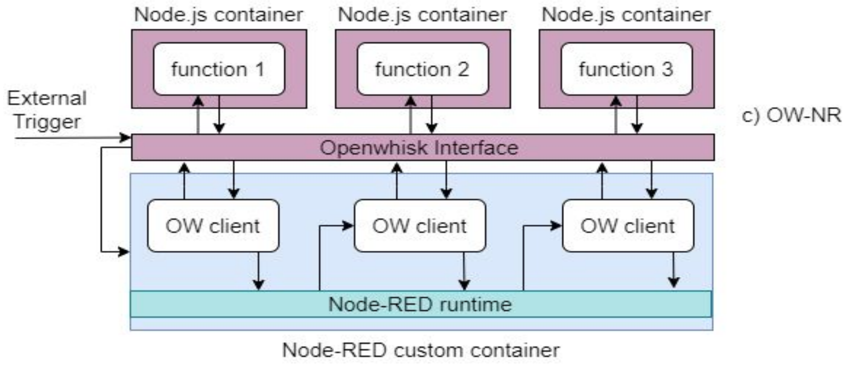
Sequences Only



Full primitives/Intra-container Parallelization



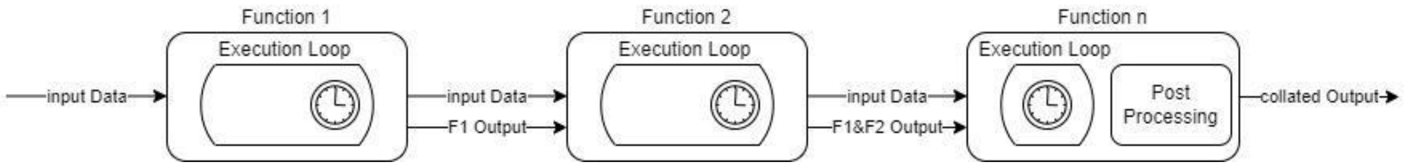
Full primitives/Full parallelization

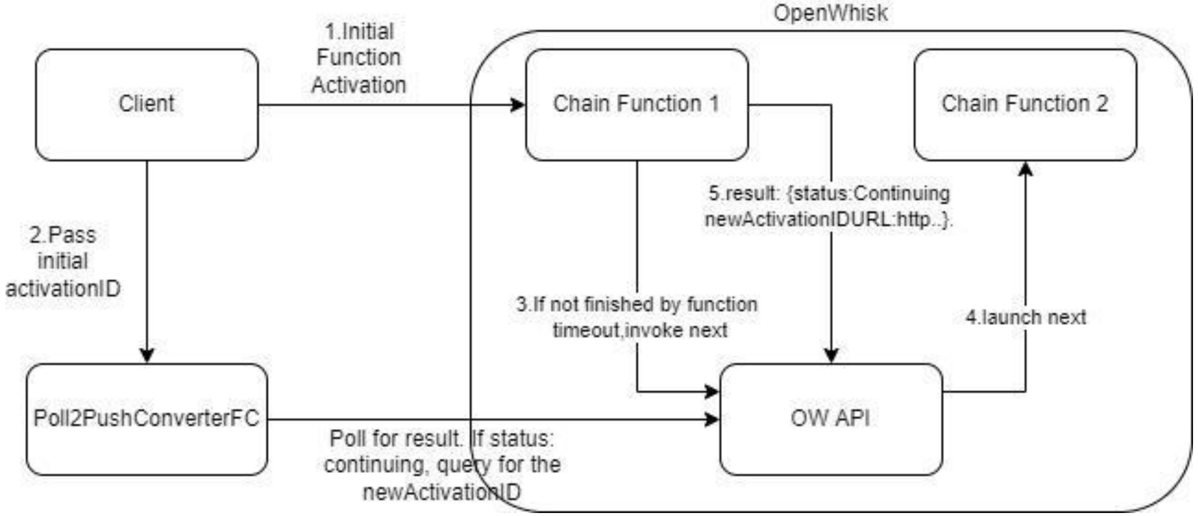


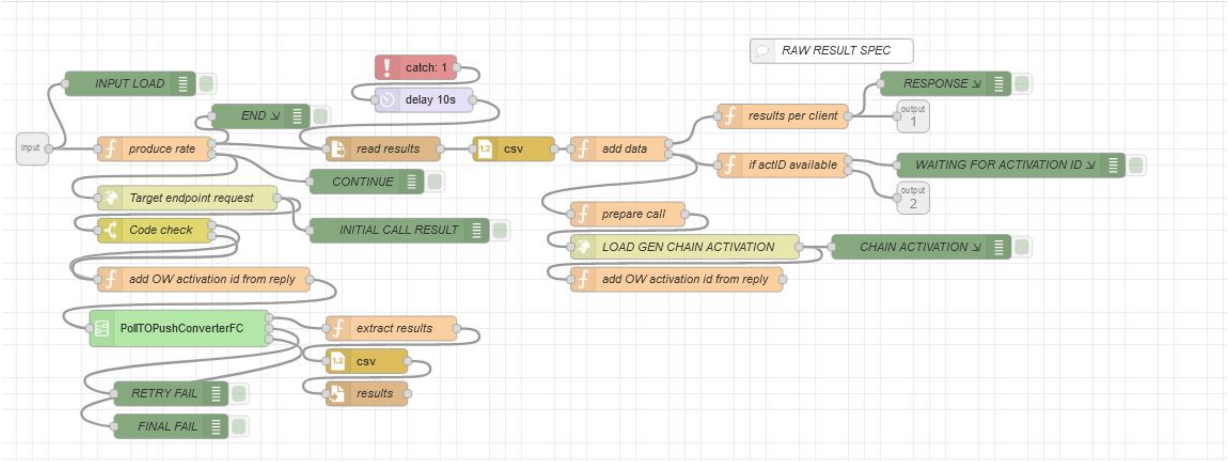
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

PHYSICS Set Rate Load Generator

- Node-RED subflow
  - Can be executed as a function
- Supports function chaining
  - To bypass OW timeout
- ~15 Metrics extracted

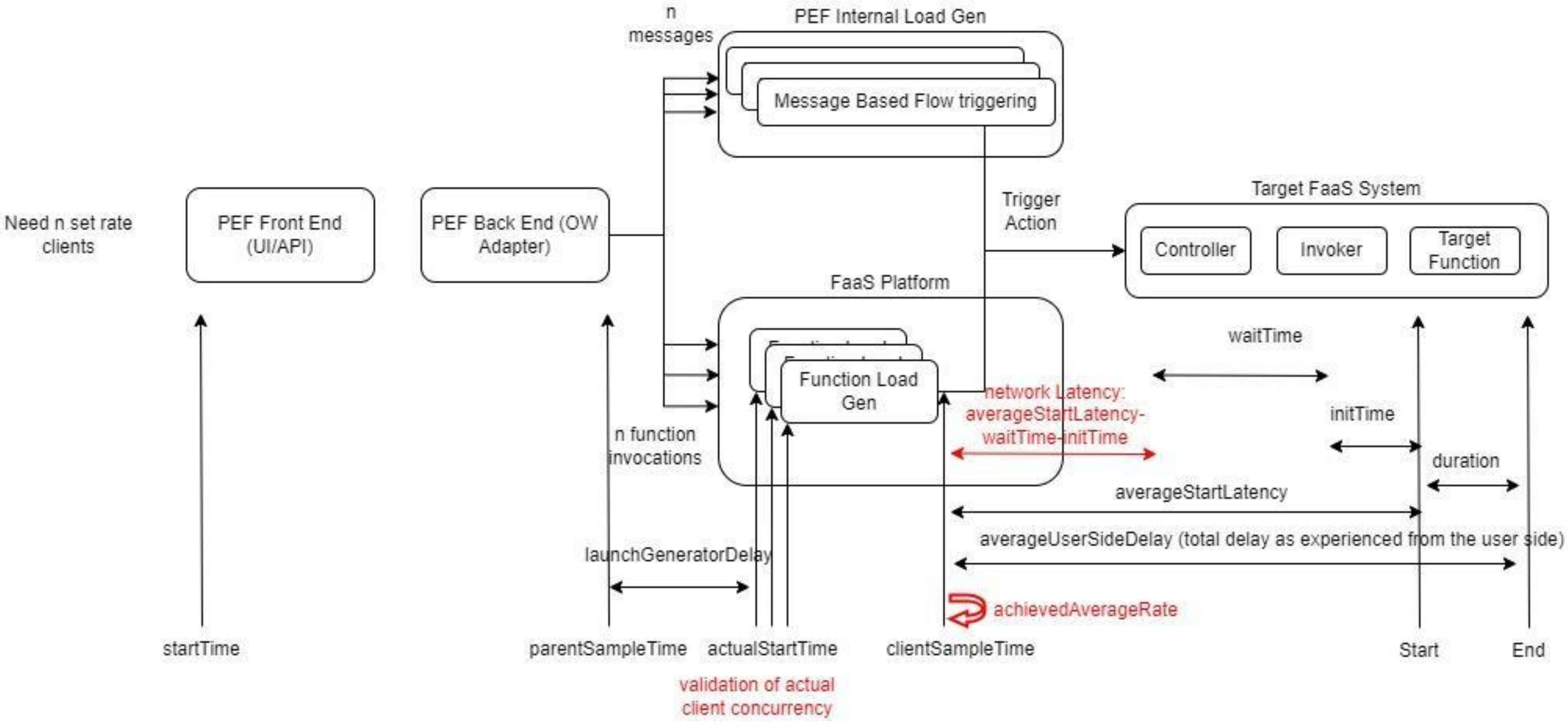






This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

PHYSICS Load Generator Metrics

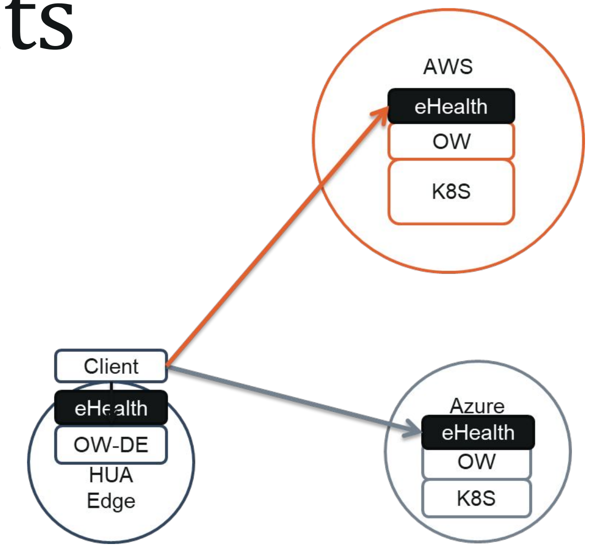


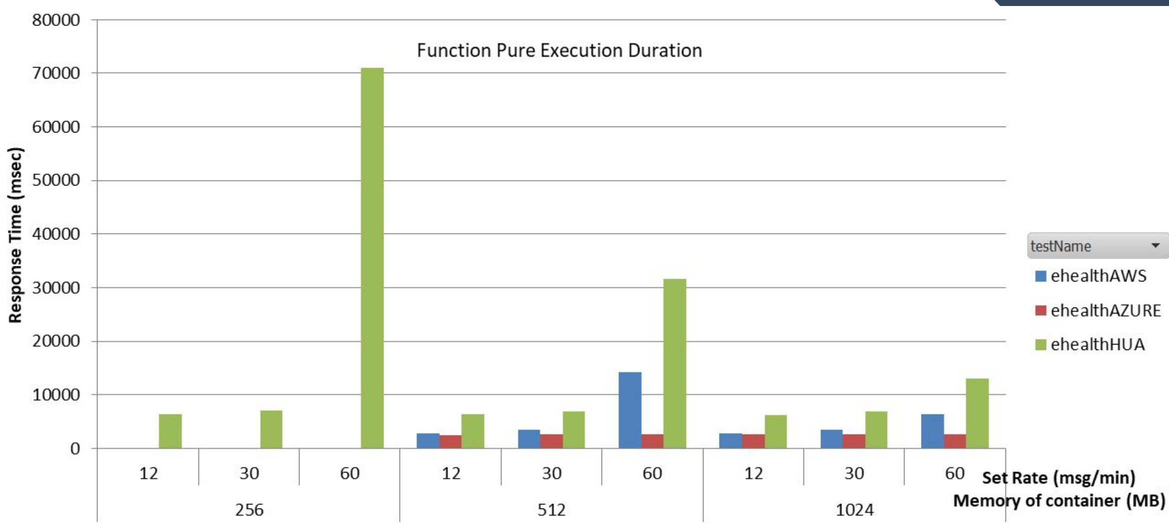
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

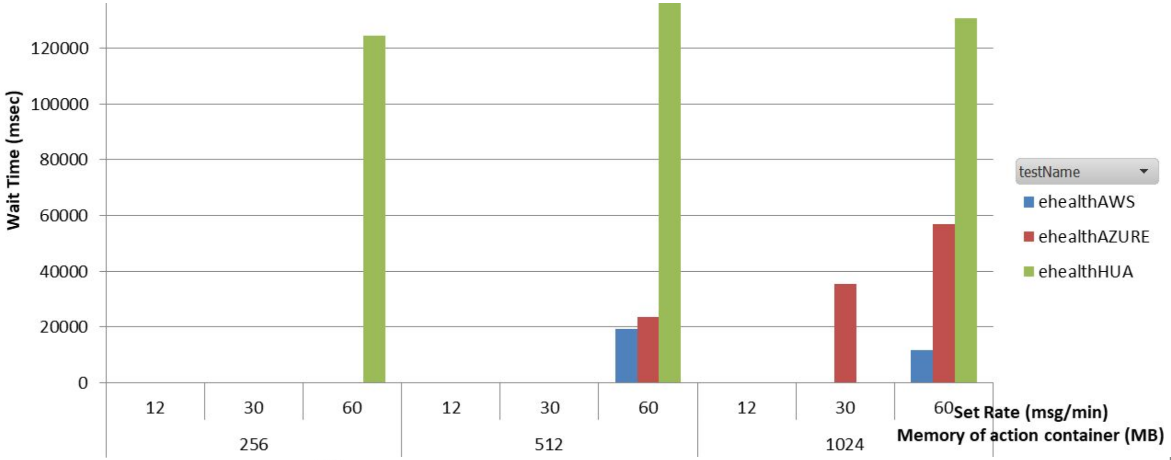
PHYSICS

Space time continuum measurements

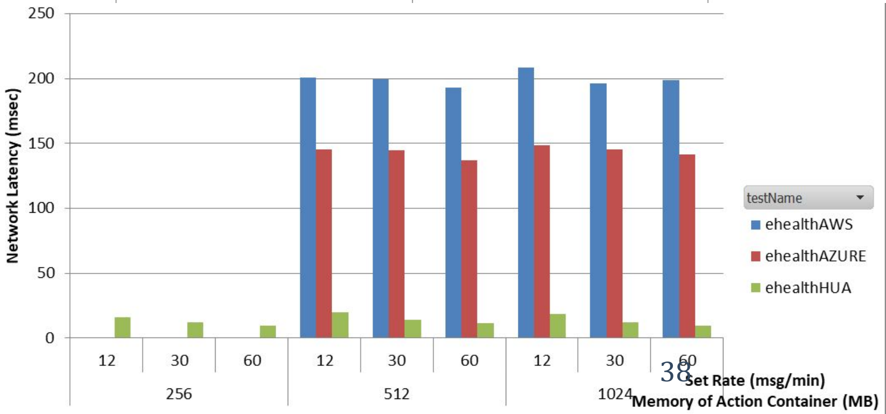
- Identification of
  - Function sizing considerations
  - Main cluster features
    - As interpreted from relative execution time, wait time, latency
  - Missconfigurations
  - Concurrent containers effect







Cluster	#workers	Worker Size	Container Pool Memory per Node	Location
HUA	1	4 vCPU-8 GB RAM	8192 MB	Greece
AWS	4	4 vCPU-16 GB RAM	8192 MB	Sweden
Azure	3	8 vCPU-32 GB RAM	2048 MB	Netherlands



ed funding from the European Unio
n programme under grant agreeme

Red Hat Role and Main Technologies

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Red Hat Main Role

- Infrastructure
  - Lower layers of the stack
    - Selection of infrastructure components: K8s, multicluster orchestration & networking
  - Infrastructure and APIs extensions
- Integration
  - API definitions (CRDs)
  - Kubernetes Operators
  - Kubernetes Webhooks

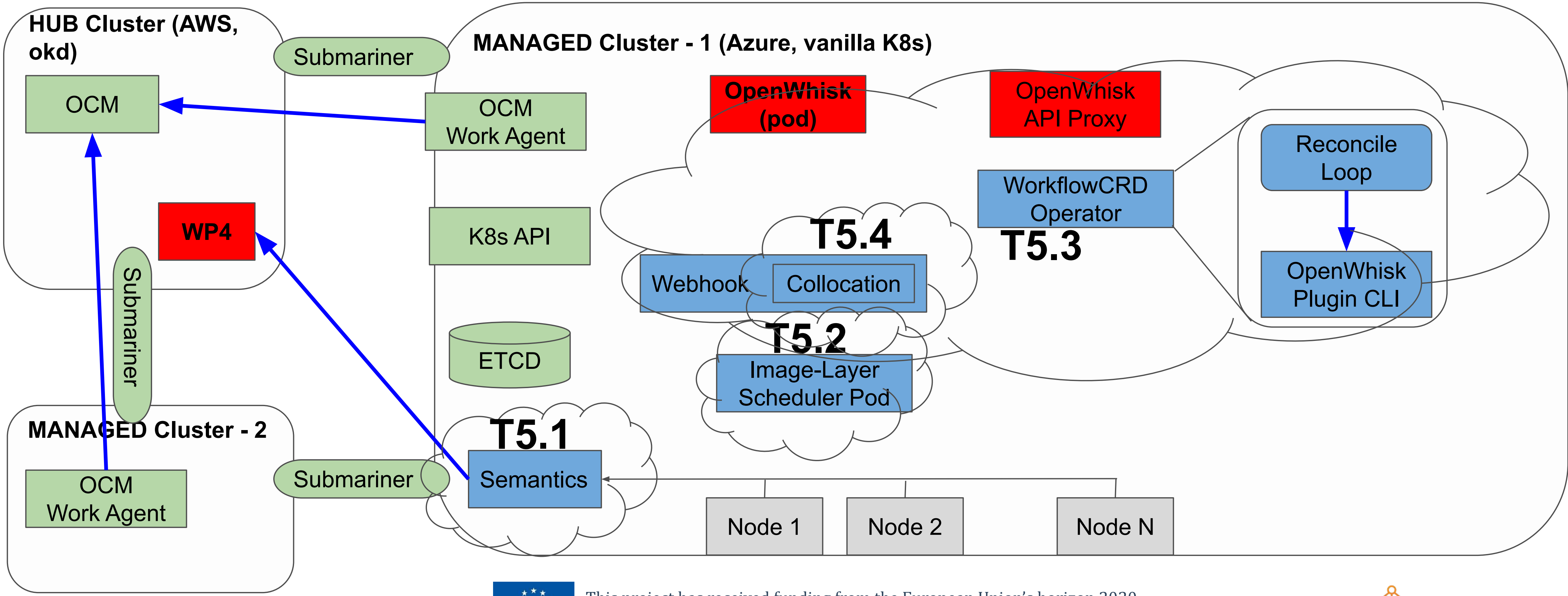
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Technologies

- Kubernetes as the core (also OKD, OpenShift Community version)
- Submariner and Open Cluster Management (OCM) for multicluster setup
- MicroShift for low-footprint k8s devices
- Prometheus for monitoring data (both infrastructure and applications)
- OpenWhisk for Function as a Service
- K8s Operators and Webhooks
  - Kubernetes way to manage applications/components running on top of Kubernetes

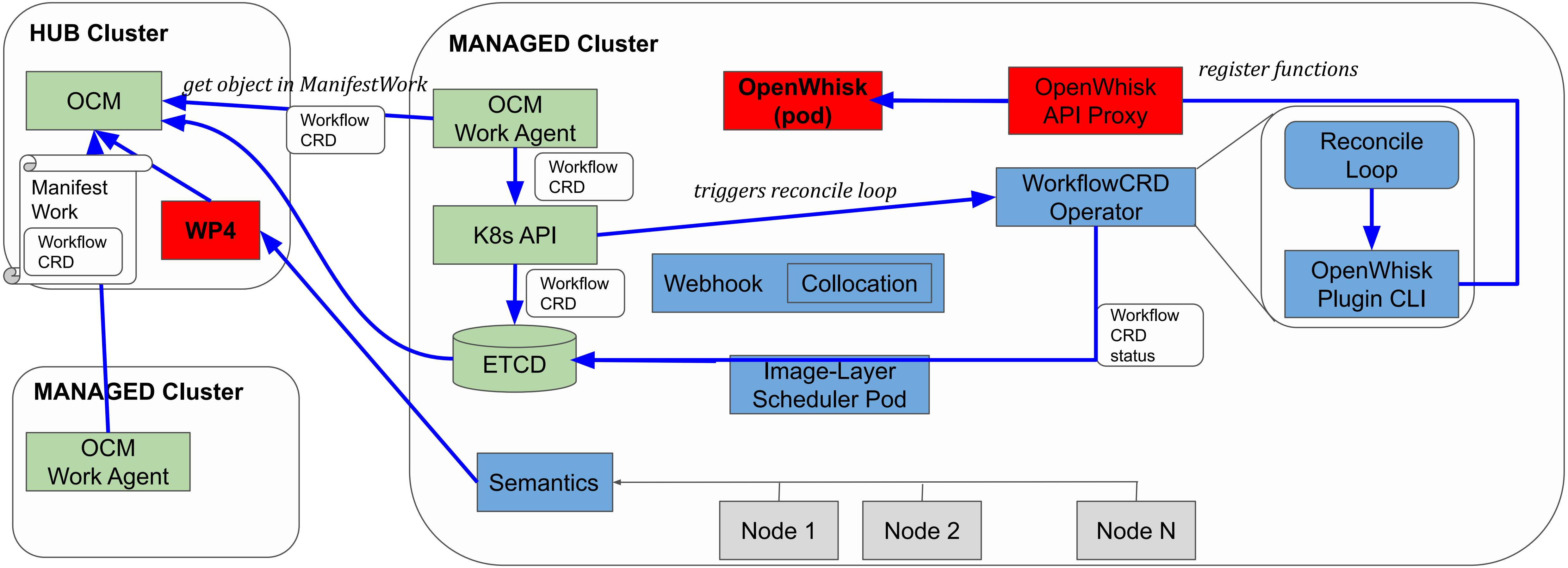
This project has received funding from the European Union's horizon 2020
research and innovation programme under grant agreement no 101017047

Components architecture



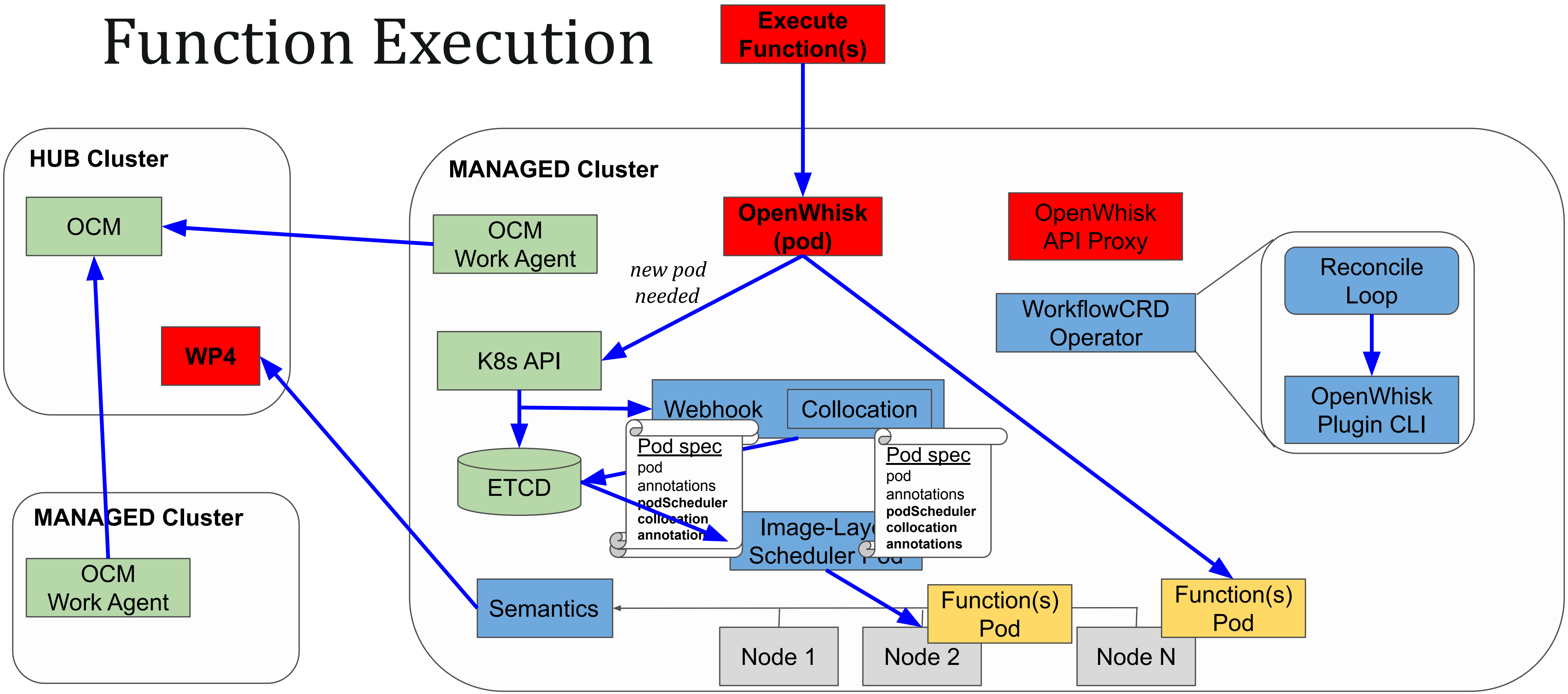
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Function Registration



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Function Execution



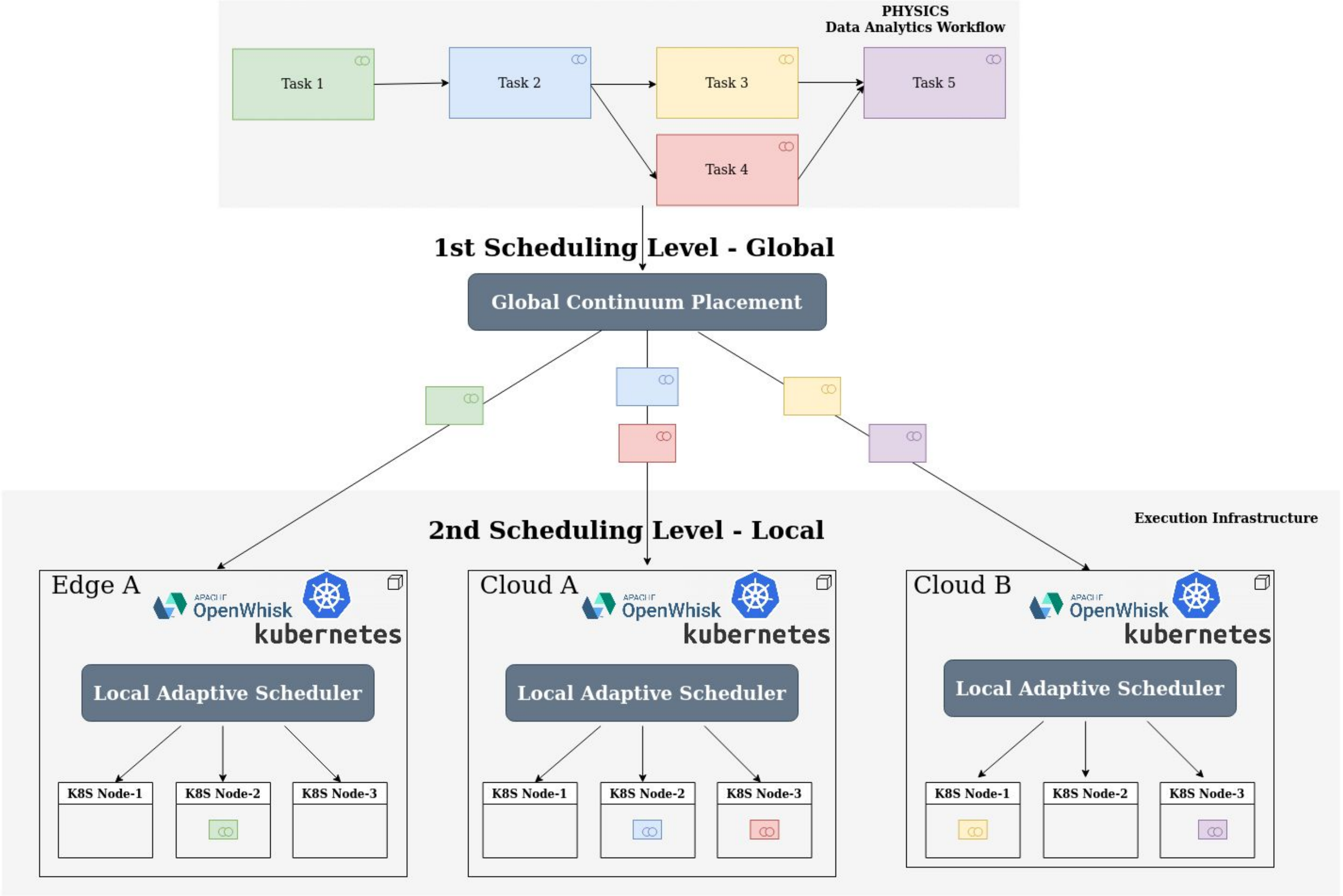
This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

K8s Image Layer Scheduling: Benefits of Open Source

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Two-level scheduling for the continuum

- Proposing a two-level scheduling procedure for the placement of the application components to available and suitable candidate cloud and Edge resources:
  - **1st Scheduling Level - Global** allows the selection of most adapted cluster considering aspects such as performance, energy, etc
  - **2nd Scheduling Level - Local** enables the selection of most adapted node considering FaaS related optimizations, etc

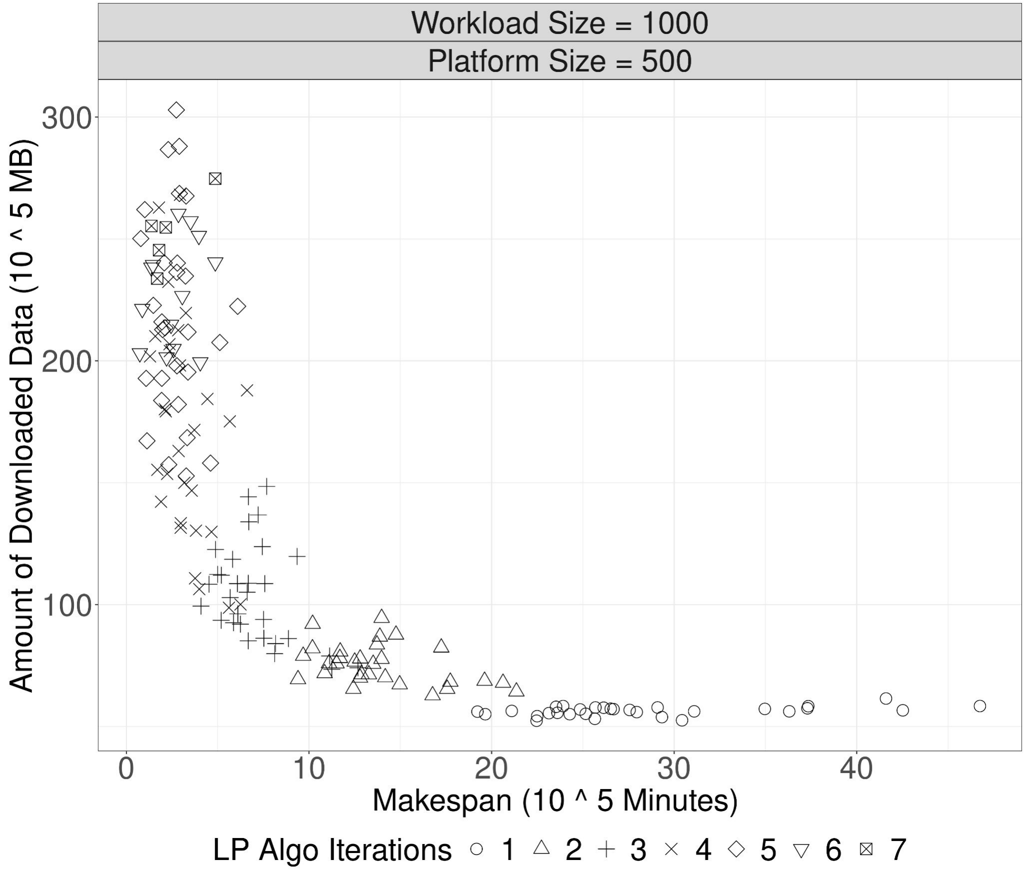


This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

1st level - Global Continuum Placement

- Implemented our first version of Global Continuum Placement component:
  - as a multi-objective scheduler allowing multiple objectives such as: Performance, Energy, etc
  - allowing the expression of both constraints and resources needs
  - the code is published as open-source: https://github.com/RyaxTech/global-continuum-placement

- Currently working on multi-objective optimizations based on **linear programming**
  - trying to minimize **makespan** and the **amount of downloaded** data by containers and function's I/O while considering platform heterogeneity
  - adaptable to various objectives
  - an article is being prepared to be submitted to CCGrid-2023

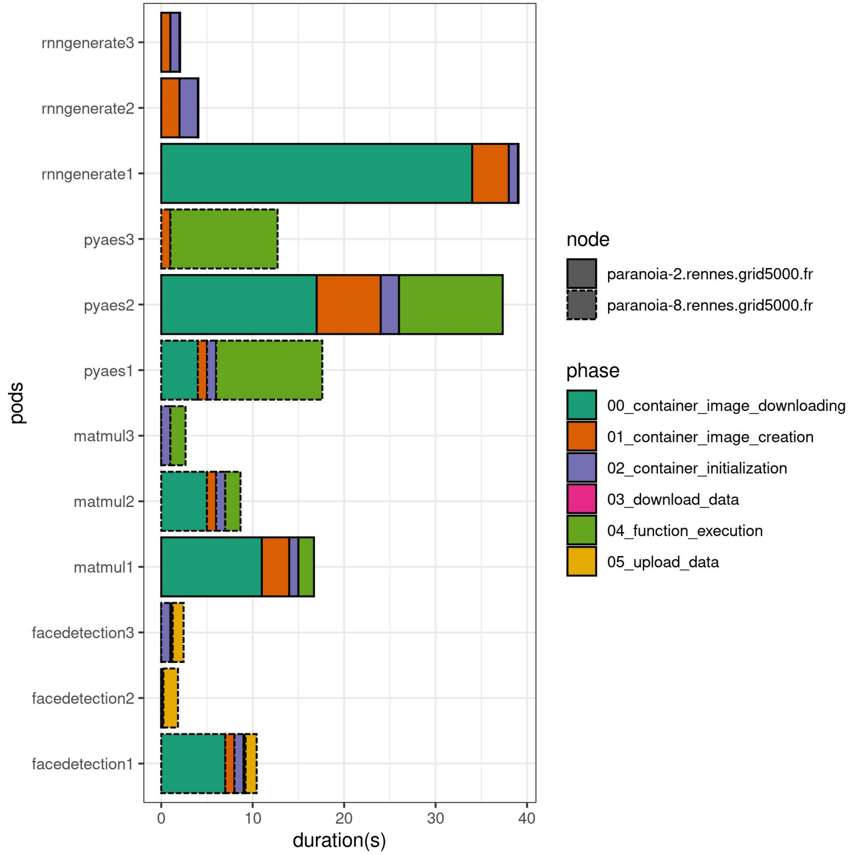


This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

2nd level - Local Adaptive Scheduling

- Focus in intelligent scheduling algorithms for efficient resource sharing (CPU, memory, storage, network), load balancing and managing resources for FaaS execution.

- Based on the results of our studies using simulations and observations
  - Our first algorithm aims to minimize the delays due to image downloading for function execution: **minimizing Cold starts**
  - Kubernetes already provides an ImageLocality plugin
  - We have implemented a variation of ImageLocality taking into account the existence of Containers' Layers and trying to favor the execution of functions on nodes where layers of the containers to be deployed exist already. The new scheduler is named **LayersLocality**



This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Layers Locality Scheduling Implementation

- For this implementation we need the following:
  - To get the available layers on each node (name and size)
  - For each new pod compute a score per node considering the cumulative size of already available layers

- For this we had to make changes on various areas of the different involved software without breaking retro-compatibility:
  - Kubernetes internal interfaces to add the Layers info into Kubernetes: optional Layers field in the core.v1.Node.NodeStatus API
  - Container Runtime CRI-O adapted to get available layers name and size on node and send it through annotations (no API change)

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

Towards pushing Layers Locality Scheduling in upstream K8S

- Open-source code and installation documentation available online on github
  https://github.com/RyaxTech/k8s-container-layer-locality
  https://github.com/RyaxTech/kubernetes
  https://github.com/RyaxTech/cri-o
- Discussions have been started with “sig-scheduling” Kubernetes group to push the new “Layers Locality” scheduler to the upstream Kubernetes.
- Currently working on experiments to compare the new scheduler with the typical Kubernetes scheduler and show the performance improvement for FaaS applications.

This project has received funding from the European Union’s horizon 2020 research and innovation programme under grant agreement no 101017047

PHYSICS

Thank you for the attention!

www.physics-faas.eu

linkedin.com/company/physicsh2020

https://twitter.com/H2020Physics

info@physics-faas.eu

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047